

PC-8801ユーザーのアプリケーションブック

No.2 Oh!88

特集

AVフルパワー! FA・MA・VA

春が来るまでに・充実プログラム12本

●全機種対応 MS-DOSの環境をディスクBASICで

COSMOS

ギャンブルをパソコンで楽しむ

ポートレース

●VA専用 斜めスクロールスーパーグラフィックスシューティングゲーム

AZON-VA

FAのサウンド機能を追究する●FA/MAの拡張モード

PCM音源ファイリングシステム●FA/MAのワープロ機能を検証する

バックスペースキー親指変換システム●ビデオアートボードを活用する

Oh!PC臨時増刊 定価980円



CMD・BACH管弦楽
カセットテープデータベース
住所録

ペンシルウオーズVA版
通信データコンバータ
etc.

言葉を究める。グラフィックを制覇する。

初代8801からPC-8801MAまで、すべてのPC-8800シリーズで動作する、パーソナルプリンティングプロセッサ「ピーワン」は、88シリーズのために生まれてきたソフトウェアです。

PC-8800シリーズ全機種対応

■日本語処理メカニズム
一枚の辞書ディスクに
すべて集約した。逐次自
動変換も、確定漢字再
変換も、選りすぐったアル
ゴリズムでコンパクトに実
現しました。

▼逐次自動変換システム AXS

サブ CPU を使用した漢字
変換システム AXS。
付属語情報の学
習機能が変換効
率をより高めます。
これで高速、高ビ
ット率の変換と豊富
な編集機能を両立
させました。

■図形をリアルタ
イム処理。文字と
絵を同時に表示、同時に編集。

▼図形編集機能

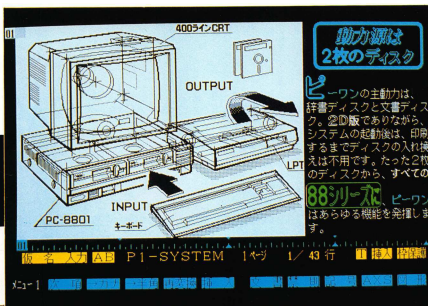
任意の位置に、任意のサイズで
グラフィックを描く。文書中のど
この領域でも作画できます。

▼グラフィック外字

テキスト画面上にグラフィックを混
在させる。16×20ドットの特異外
字を発想できた時、この難問が
解決しました。図形編集時はピク
トイメージデータ、文字編集時は
キャラクタ、目的にあわせてデー
タ型を自動変換します。

▼8種類の文字装飾

文字を加工する。縁どり、反転、
強調、回転、鏡文字、囲み、斜体、
カラー指定が一文字単位で指定
できます。しかも、加工結果をその
まま画面表示。複数の装飾を同時
にけることもできます。



▼確定漢字再変換機能

確定済みの漢文字列を未確定ひらがなに復帰させる。一定量の入力文字列をバッファリングしておくことにより、校正作業の効率が飛躍的に向上しています。

標準を発揮します。

きのうを発揮します。

機能を発揮します。

反転文字

斜体文字

縁取り文字

強調文字

カラー文字

囲み文字

回転文字

鏡文字

88、ワープロ、ピーワン。



ワードプロセッサの機能は

どこまで高度化しうるのだろうか。

漢字変換効率、文字種、印刷精度の向上。

さまざまな問題が次々に解決される中、

文字と絵を自由に扱うことが

大きな問題となって残された。

パーソナルパブリッシングプロセッサ「ピーワン」。

たゆまぬ高機能化の要求に対する

デービーソフトの一つの回答が、

ワープロの世界を大きくひろげていく。

好評発売中

5.25D標準価格
40,000円
(対応機種)
PC-8800全シリーズ
& PC-88VA



■フォントディスク

印刷に関する処理はすべてフォントディスクが行なう。原稿用紙やはがき印字にも対応しており、24ドット高品位フォント(明朝体、ゴシック体)を使用した美しい印字結果を得ることができます。



用紙サイズと印字範囲の指定を視覚的に表示します。

■グラフィックエディタ

200ラインカラーと400ライン白黒の二つのモードを持ち、ピーワンの図形編集機能とデーク互換がある。グラフィックツールとしてより高度なグラフィック編集機能を充実させました。



先進のカスタマイズ機能

■装飾パターンの任意編集

網かけや下線、枠線、外字などをマウスを使って簡単編集。ROMの漢字を読みだしてオリジナルの外字に編集し直すこともできます。

■装飾パターンのファイル化

自分で編集した網かけなどのパターンをファイル形式で登録。使いたい装飾パターンを選択できます。同一文書で装飾パターンだけ変更して印刷することも簡単です。

■操作キーの任意設定

変換キーやコントロールキーによる機能を再割り当て。メニューやスケールの表示色を変更する。自分が使い易い、自分だけのピーワンを自由に設計できます。

▼ピーワン2HD版ご希望の旨へ

2HD版ご希望の方で、2D版をすでにご購入の方には2HD版発売以降にメディア交換サービスを準備致しております。(当サービスの対象は2D版をご購入のうえ、同梱の「登録申込書」を弊社までご送付して頂いた登録ユーザーの方に限らせて頂きます。)

88通信ソフトの決定版



初心者でもわかりやすい操作性
ベテランでも満足する高機能

これだけで通信ができます。
5.25D版 標準価格 12,800円
PC-8800シリーズ & 88VA 対応

ピーワントームの主な仕様

- 通信画面がスクロールバック
- 19行×25列のバッファで380行まで
- 文書編集機能拡張
- 通信と共用のバッファ使用でメール作成などにも最適
- 漢字コード自動判別
- シフトJIS、NECJIS、新JISに対応
- ピーワンの日本語処理を簡単操作性、辞書レベルで互換性あり
- 全角判読可能なオートバリエット専用のエディタを装備。主要BBS用サンプルプログラム付き
- MODEM方式の送受信が可能
- ピーワン、BASICなどのファイルを直接使用
- ピーワン、希望シリーズ、BASICのファイルをコンバート
- 通信中でも使用可能なユーティリティ
- データディスクフォーマットなど

ピーワンとデータ互換の世界をひろげる



他社8ビットワープロの文書ファイルをピーワンに
コンバート 5.25D版 標準価格 6,000円

ユーカラK2/Art/mini・JET-8801A/V2・即戦力・SHOGUNの文書をピーワンの文書にコンバートする事ができます。



文字サイズや装飾、枠線をコンバートできます。
(一部コンバートできないものもあります。)

※各ソフトは各社の標準です。

FH/MHの上位機種、サウンドマシンとしてのFA/MA。その新しい機能、FM音源6音、ADPCMを余すことなく徹底追究。また、VAのビデオアートボードの活用方法A to Zと合わせて、88の繰り広げるAVの世界を解説。

特集

AVフルパワー! FA・MA・VA

PART.1 FA/MAのサウンド機能を追究する 志村隆人……………17

- Sec.1 CHECK IN サウンド機能 FA/MA
- Sec.2 サンプリングプログラム STUDIO 88
- Sec.3 FM音源を学ぶ
- Sec.4 ミュージシャンライクに インスタントミュージック

PART.2 FA/MAの拡張モード 坂東浩幸……………33

- Sec.1 FA/MAのBASIC
- Sec.2 PCM音源ファイリングシステム「VOICE MEMO 88」

PART.3 FA/MAのワープロ機能を検証する 斉藤研二……………45

- Sec.1 FA/MAの日本語処理能力を探る
- Sec.2 バックスペースキーを親指で「親指・BACKSPACE」

PART.4 VAのUp Grade Board 石葉直志……………51

ビデオアートボードを活用する

スーパーパソコン88VA I

アーティストのツールとなった88VA 桑原三知子……………8

スーパーパソコン88VA II

商品開発のプレゼンテーションに使用された88VA 沢井邦仁……………10

FA/MAを分解する……………12

ビデオアートボードを活用する VAあーと・アート 石葉直志……………14

究極の超初心者のために

最後のマ・シン・語—考え方を考えよう 田奈真実……………55

88VAの救世主

What's「MS-Engine」? 佐々木誠二……………60

88 GRADE UP TOOL 後藤清治……………64

オムロン ハンディイメージスキャナ「HS10R」

読み物

・UNIXは米国ベル研究所で開発されたソフトウェアです。また、オペレーティングシステムCP/M, CP/M86はデジタルリサーチXENIX, MS-DOSはマイクロソフトの登録商標です。その他プログラム名、システム名、CPU名は一般に

各メーカーの登録商標です。本誌に掲載されたすべてのプログラムは著作権法上個人で使用する目的以外で無断で複製することを禁じられています。各記事は対応機種(メモリ)、MS-DOSのバージョン、必要周辺機器などを確認してご利用ください。

読み物

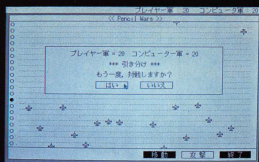
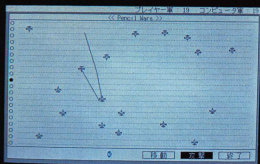
68	IMAGE TURBO 88	猪久間達也
72	「プリンタ」	木内 等
78	88TERMを使いこなす	山本幸子
152	「リーズナブルな道へモテム」	
81	COSMOS	篠田和義
95	NICE ACCESS	古村勇一
97	BACH管弦楽デモンストレーション	小林大二
112	AZON-VA	大島 麓
126	Cassette Manager	村田晩生
131	ポートルース	米田稔一
141	MANDATA	佐藤 淳
146	PENCIL WARS 88VA版	竹田津 恩

アプリケーション

2	目次
4	88 WINDOW
7	特集とびら
157	読者のページVOW WOW
159	Oh!88 INFORMATION
160	編集後記



表紙イラスト
藤枝リュウジ



88VA版Pencil Warsの登場です。マウスさばきも軽やかに、行くぞ我らがF-16。「前方に敵機発見 / これから迎撃に向います」「了解」なんてやっているうちに、敵機からも攻撃が / 紙一重で回避して必殺の一弾を / と思ったら、あらハズレ。

Pencil Warsの操作は、すべてマウスで行います。マウスを動かしやすいように、邪魔なキーボードは片付けておきましょう。ゲームを終了するには、(終了)を選択。コンピュータ軍をせん滅するまではやめられまへん。

懐かしのエンビツ飛ばし

PENCIL WARS 88VA版

146ページ

斜めスクロール
スーパーグラフィックス
シューティングゲーム

AZON-VA

112ページ

中央の白い機体がプレイヤー機。対空ミサイルと、対地プラスターを装備。88VA 標準装備のマウスか、市販のジョイスティックで操れ /



この機体はミサイル攻撃してくるので要注意だ。動きは単純なので、早目に撃破しよう (実物では機体は赤に変更)。

左右に旋回しながら編隊でやって来る戦闘機。プレイヤー機の直前にある赤玉は敵の追尾ミサイルだが、これを振り切るのは大変。

AZON-VA

どうですか画面の美しさ。キャラクターの素早い動き。そしてスムーズな斜めスクロール。信じられないかもしれませんが、これが何と、BASIC オンリーで作られたゲームなのです。



敵の攻撃が止んだすきに地上物を破壊／ 攻撃してくる建造物は早目にプラスターで攻撃するのが得策。



後半になると、いろいろな飛行パターンの敵が同時に出現するので避けるのが大変。後方から襲ってくるやつもあるし、しつこく追いかけくるものもある。

エリアの最後のシヤン。ここを越えられれば1ラウンドクリアーだ。開発中バージョンなので物しそに見えるが、実際には熾烈な攻撃が襲ってくるぞ／



AVフルパワー! FA・MA・VA



88がおもしろくなってきた。VA が登場したのが今年の春。16ビット CPU、PC-Engine 搭載による65,000色のグラフィック機能にド肝を抜かれ、何やら得体の知らない大物に88ユーザーはドキドキさせられた。そして、びっくりしている間にこんどは「音」で勝負する88が飛び出してきた。その名はFA/MA。FM 音源が6音源、その上あろうことにもサンプリング機能を搭載しているのである。VA のグラフィックスを目のあたりに見たときの驚きを、また体感させられたユーザーも多いだろう。VA でアニメも描ける、ビデオも取り込める。FA/MA でダイナミックスレンジが楽しめる、作曲もできる、サンプリングもできる。こうなりゃ、新しい88は AV を感じずにいられない。ドキドキ、ワクワクこれからの88が楽しみ！

スーパーパソコン88VA.I

アーティストの ツールとなった 88VA

—62年9月20日～9月23日
東京芸術大学「芸術祭」における
「電子道具箱カンパニー」の4日間—

スタジオBC
桑原三知子

88VAがアーティストたちの手にかかる
とこんな力を発揮する。昨年、9月に開催
された東京芸術大学(上野)の芸術祭で「電
子道具箱カンパニー」というCGグループ
が小さなブースを開いた。彼らはアートから
コンピュータへのアプローチをしている
芸術家の卵たちである。彼らの手にかかる
と65,536色のVAのグラフィックスは、こ
んなにもすばらしい。

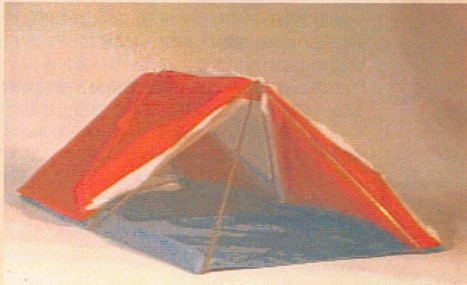


芸術祭での電子道具箱カンパニーのブース風景



オールナイトパーティー風景

88VAで描いたオールナイトパーティーのチラシ・
モノロー版(上)。88VAで描いたオールナイトパ
ーティーのチラシ・美女版(中)。立体作品のプレ
ゼンテーションに使用した絵(下)。



「Z's STAFF」を使用して
描かれた作品

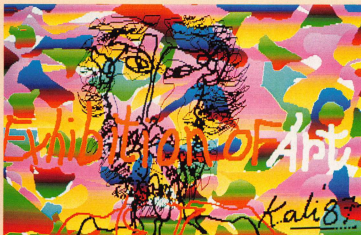


上と同じソフトを使用して
描かれたネコ



上と同じアーティストによ
るネコ

体験コーナーで外国人留
生によって描かれた作品



「電子道具箱カンパニー」は、東京芸術大学の学生を中心とした、コンピュータグラフィックスやライティングアートに興味のある有志十数名からなるグループです。

このグループでは、88VA が発売されてからCGによるいろいろな作品を制作してきましたが、昨年9月に行われた芸術祭でその作品を発表することになりました。

芸術祭が開催された場内では、アトリエを画廊に見立てたグループ展が数十か所で開催され、電子道具箱カンパニーもこのブースとして参加することになったわけです。

問題の88VAを初めてとするマシン類は、NECの方にお願いいただき、そのお礼として最終日に行われるレーザーショーでNECのロゴを流すことにしました。

これらの準備と作品の制作は、同時進行で行われたわけですが、マシンの準備が終わったあとは、いっそう作品の制作に力が入り、毎晩夜中まで制作するという毎日が始まりました。

そして芸術祭前日、いよいよ会場作りです。会場には、電子道具箱カンパニーの作品展示のモニター3台以外にも、展示を見にいらしたお客様に実際に88VAに触れて遊んでいただくコーナーや、オールナイトパーティーのチラシをプリントアウトして配るコーナーも設け、お客様にも88VAで実際に楽しめるように心がけました。

私たちのグループの作品はアニメプレーマーを使ったアニメーションと、「Z's STAFF」を使った静止画を中心としたもので、それをプリントアウトして会場に展示しました。

芸術祭当日、スタッフの心配をよそに会場には続々と人が集り、88VAのまわりはあっという間に満員御礼となりました。

では、実際に88VAで遊んだ人たちの感想は?

「絵具で描くのと同じくらいよく描ける」

「ビデオとのデジタイズがおもしろい」

「色がたくさん使えて、とてもきれい」

アニメプレーマーに関しては、

「思ったより簡単に絵が動く」

「素朴なアニメーションが使っておもしろい」

アニメプレーマーはじっくり時間をかけて楽しんでいる人が多かったようです。

最終日のオールナイトパーティーでは、壁に映し出されたNECの文字がパーティーの最後を飾っていました。

アートの世界では今ではコンピュータは1つのツールとして位置づけられています。芸術においては、まだパソコンユーザーはひとにぎりにしかありませんが、今後、私たち電子道具箱カンパニーでは、88VAのアートの世界を追究し、VA独自の世界を広げて行きたいと思っています。

スーパーパソコン

88VA.Ⅰ

スタジオBC 沢井邦仁

スーパーパソコン88VA がその65,536色のグラフィックス表示能力を買われ、これまでのパソコンにない分野で注目されてきています。ここで紹介するのは、ある商品のプレゼンテーションに使用された例です。クリエイターの手にかかった88VAのすばらしいグラフィックスをご覧ください。

■「プレゼンテーション」とは

「プレゼンテーション」とは、商品の開発などの場で、ものを相手に伝えるための手段です。その方法と種類や規模はさまざま、また分野によっても異なっています。たとえば、商品パッケージのデザインを依頼されたデザイナーは、自分のアイデアを相手に納得させるために、パネルに必要な設計図、デザインコンセプト、図案、パリエーション、商品ディスプレイのシミュレートまで行い、さらに口頭で解説し、プレゼンテーションしていきます。そのツールとしてコンピュータグラフィックスが使用されることがよくあります。

今回、芸術祭での作品展示にVAを使いプレゼンテーションを行いました。その中で内容とCGのメリットを写真と合わせて解説していきます。

■プレゼンテーションにおけるCGのメリット

プレゼンテーションで大切なのはそのコンセプトを伝えることです。この電話機のコンセプトの場合、「未来的なイメージを持ち、なおかつ馴染みのある造形を有し、インテリアとしてもその存在が生きるデザイン」という感じで進めました。要するに電話機のうり文句です。また、その説明の補助に、イメージをふくらませるための写真や、使う人のターゲットを資料としてつけて加えていきます。これらは、本来ならばパネル上に文章や表、フォトイメージを並べて構成するのですが、コンピュータを使い、モニタ上でそれらを展開することでよりわかりやすくなりました。たとえば項目別にファイルを用意し、解説と見出しにそのファイルを開いていけば、演出効果も出るというもの。このような方法はビデオやスライドによるものと似ていますが、ビデオは受動的で、スライドでは即時性に欠けます。

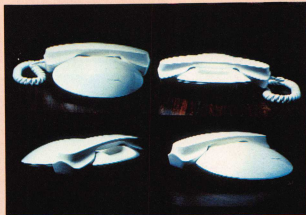
次にデザインに關してのプレゼンテーション。本来であれば、造形を説明するのにドローイング（製図）、レンダリング（注）などをやはりパネルに展開していきますが、それぞれがすべてで行われるために、イメージが固まるのに時間と手間がかかりま

す。その点コンピュータを用いた場合、図面の段階での手間を除けばレンダリングはほぼ自動的に行え、しかもVAの場合65,536色を駆使すると、モデルイメージがはっきりと浮びあがります。次にカラーリングですが、この場合も1度画像が生成されればそれを基に色を変えてモデル作成前のシミュレーションが十分に行えます。この場合も、紙の上では下絵を基にデザイナーが勘をたよりに何十枚もの提案をし、その中から選びぬいていく作業をしますが、コンピュータでは、リアルタイムにデザイナーの意思が反映されます。また、デザイン検討の段階で難しい機能のシミュレートも、CGの架空イメージでそれを伝えることができます。この電話の場合、着信コールと同時に光によるインジケートを加え、演出することを考えたため、そのシミュレートにもVAを用いて速くリアルなものができました。今回使用したVAに関しては、色のシミュレーションでは65,536色のグラフィックスのおかげで十分満足のものでしたが、設計におけるシミュレートはソフトの充実を期待したいところです。また、コンピュータのプレゼンテーションの画像は、その場でプリントアウトできるというのもメリットでしょう。

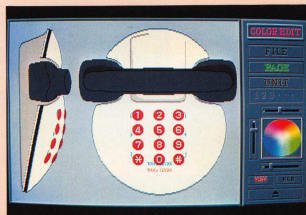
■万能ではないコンピュータ

このようにコンピュータを用いたプレゼンテーションはリアルタイムのよさと応用範囲の広さで、かなり有効な手段であるといえます。しかし、コンピュータでは伝えられない要素もまだまだ残されています。ザラザラ、つるつる、すべすべといった質感や、重さといったものは、コンピュータでは表現できません。そのような意味で、デザインとコンピュータの融合はまだ始まったばかりでしょう。しかし、大手企業ではそれらは次第に統合がなされてきて、今後は設計から意匠、製品化の過程からコンピュータが介入し、プレゼンテーションの形態も変化していくように思われます。

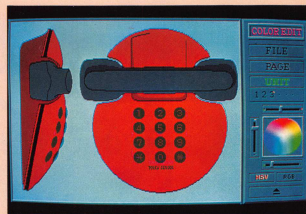
VAに代表されるパーソナルコンピュータの可能性はまだ私たちの想像をはるかに超えて広がっているような気がします。注 レンダリング……ここではラフスケッチの意味。



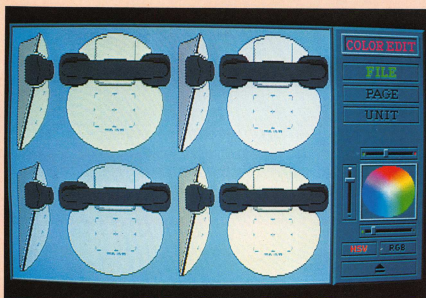
① 取り込み合成



② カラーバリエーションのシミュレート（プッシュホンタイプ）



③ カラーバリエーションのシミュレート（タッチセンサータイプ）

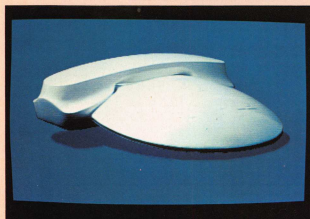
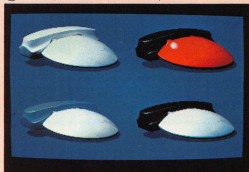


④ 微妙な色同士の比較検討

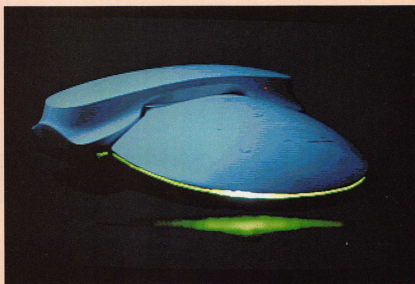
⑤ カラーシミュレート
(取り込み画像に修正)



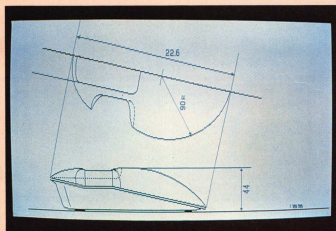
⑥ カラーシミュレート



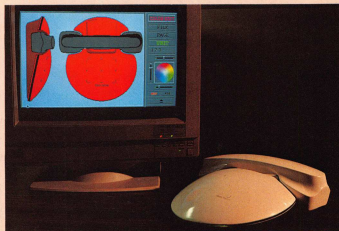
⑦ ペイントツールによる
処理



⑧ ⑦を基に光のシミュレート



⑨ ドローイング画面



⑩ 展示の状況 展示風景

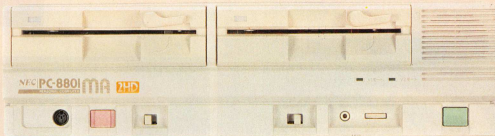
FA/MAを分解する



FH, MHの上位機種FA, MA。サウンド機能の強化と新しい辞書ROMが搭載されているが、その機能の詳しい説明は本誌の特集をご覧ください。ここではハードを分解してその構造を見てみることにする。

本体前面

ボディシェイプは、FH, MHからほとんど変わらないが、スライドスイッチなどが入っていたドアがなくなった。このなかにあったV1, V2モードの選択スイッチは、ドライブ1の下に配置された。また、ディスプレイの周波数の15Kと24Kの切り替えスイッチは、本体後部に移った。

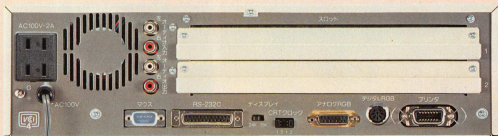


本体後部

サンプリングのためのオーディオ入力、オーディオ出力の端子を備えている点が新しい。

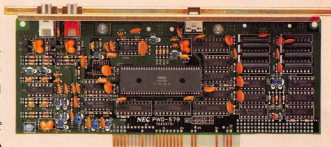
電源は、従来はプラグタイプで本体から外せたが、コストダウンのために直接コードが出ているタイプに変わった。

FAの場合、FHからの合理化は徹底しており、カセットインタフェースとともに、何とデジタルRGB出力まで廃止されてしまった。写真はMAで、RGB出力を持っている。また、FAでは拡張スロットは1個である。



「サウンドボード」(PC-8801-24)のバージョンアップ「サウンドボードⅡ」(PC-8801-23)。中央の「YM2608」がADPCMを司るヤマハの音源チップ。

その右上に4個ずつ2列に計8個並んでいるのが、サンプリング機能のためのRAM, 「D41256V-15」。



内部

メインRAMおよび拡張RAM
D41464X8 (FAではX2)
(ZIPタイプ)

高速テキストV-RAM
D4364C-12

グラフィックV-RAM
D41416-15X6

サンプリング用メモリ
D41256V-15X8
(ZIPタイプ)

音源チップYM2608

メインCPU78008AC-8

辞書および漢字フォント用
4MビットP-ROM D23C4000C

CRTコントローラ
D3301AC-Z

大規模
カスタムLSI

ノイズフィルター

インタフェース
コネクタボード

ディスクサブシステムCPU
D780C

ディスクサブシステムROM
D23C64

フロッピーディスクコントローラ
D765AC

ディスクコントロー
ルボード

8800シリーズでは、SRあたりから内部回路のカスタムチップ化などによる部品点数の減少と小型化が図られてきた。カスタムチップは（量産を前提に）コスト面でも有利である。

FA/MAの基盤を見ると、3個ほどカスタムチップが使われている。そのうち一番大きなものは、27mm角で160ピンのフラットパッケージという巨大なLSIで、かなりの集積率と思われる。これらのカスタムチップが、どのような働きをもつのかは不明だが、FA/MAの基本機能の大部分が

これらのなかに封じ込められているのだろう。

基盤上のLSI実装効率をよくして小型化を図るため、拡張RAMやサンプリング用のメモリには、ZIPパッケージのものが使われている。ZIPとは、基盤に対して垂直に立てて装着するタイプのLSIパッケージのことで、通常のゲジゲジ型のDIPパッケージに比べて、同じ面積に約2倍の数を実装可能だ。

MAは日本語変換のための256Kバイトの辞書データがROM化されているが、こ

れは何と4メガビットのマスクROM1個のなかに、JIS漢字第1第2水準のフォントとともに納められている。

各種インタフェースコネクタは、別の基盤に組まれているうえ（従来は1枚基盤）、ノイズフィルターも多数挿入されており、かなりノイズに気を遣った設計であることがわかる。

ディスクドライブとコントロール基盤との接続にシートタイプのフレキシブルケーブルが使われていることも目新しい。

1. ソフトによる作画

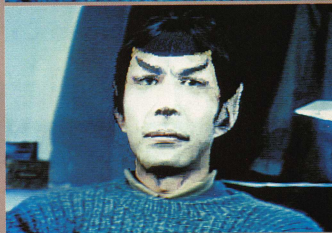
① 変身前



② 変身中



③ 変身後



2. プログラムによる作画



④ もとのデジタイズ画像

ビデオアートボードを活用する

VA
アート
あと

VAが市場に出て、約一年。「65,536色のグラフィックス」はパソコンユーザーの間では知らない人間がいないほど、浸透している。そしてパーソナルユースの分野だけではなく、コンピュータグラフィックスの世界にもVAは進出してきた。

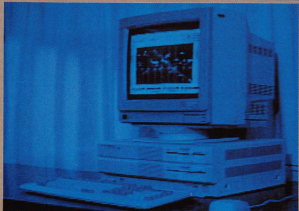
その証拠にこの春の某化粧品メーカーのプロモーションビデオにVAが使われたのである。これは、VAのグラフィック機能が、いかに精度の高いものであるかを物語っている。

ここでは、プロモーションビデオに使われたCMをはじめ、VAとビデオアートボードで作られたグラフィックスをカラーでご覧いただく。なお、ここでご紹介した画面の作成方法は本誌特集51ページからの「ビデオアートボードを活用する」で解説している。

3. ビデオカメラを使用する



7 左側メガネ用



8 右側メガネ用



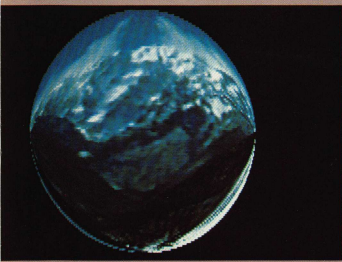
9 赤青メガネ用3D静止画像

4. そのほかの活用例

10 もとの新宿高層ビル



11 ⑩の左右2つのビルを消した想定図



5 ④をプログラムによって丸めたもの



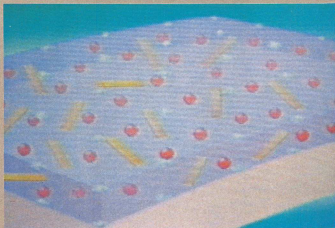
6 ④をプログラムによってゆがめたもの

スーパーインポーズ機能を使う



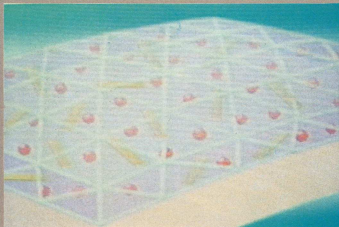
- 12 スキーのビデオに、タイトルアニメーションをつけた例

VAがCMの絵を作った!



- 13 口紅の概念図

- 14 徐々にネットが広がり、結合します



- 15 完成図

FA/MAの サウンド機能を追究する

志村隆人

- Sec.1 CHECK IN サウンド機能 FA/MA
 Sec.2 サンプリングプログラム STUDIO 88
 Sec.3 FM 音源を学ぶ
 Sec.4 ミュージシャンライクにインスタントミュージック

PC-8801FA/MA が88シリーズの仲間入りをした。その機能のなかでも最も注目されているのがサンプリング機能を搭載したサウンド機能である。パソコンで気軽に本格的なコンピュータ・ミュージックが楽しめるようになった。そこで PART.1 ではサウンドマシンとしてのFA/MAを徹底追求してみよう。

Sec.1

CHECK IN サウンド機能 FA/MA

FA/MAの OPNAとは?

PC8801-FA/MA および PC-8800 シリーズに作られた新サウンドボードには、これまで一般に使われてきた FM 音源チップ OPN (YM2203) とは異なり、さらにグレードアップした新チップ OPNA (YM2608) が搭載されている。

ここではその新シンセサイザチップの機能および使い方について詳しく解説をしていきます。

FM音源チップを ふり返る

この OPNA について説明する前に、現在市場に出回っているその他の FM 音源

IC について説明をします。

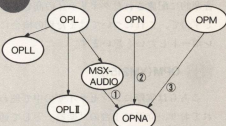
なぜ、最初にこのような説明が必要なのかというと、今回紹介する OPNA はそれら FM 音源チップのよいところばかりを集めた IC であるからです。

OPN(YM2203)

これはもうパーソナルコンピュータの世界ではスタンダードになっている「元祖」FM 音源チップです。PC-8801mkIISR 以降の機種には標準装備され、またその他の機種のパーソナルコンピュータにも搭載されており、もう知らない人はいないというほど有名なチップです。

内部には4オペレータ構成の FM 音源部が3ボイスと SSG 音源がドッキングして入っています。

図1 FM音源チップの相関図



OPNAは種々のFM音源チップからオイシイところを取って作られている

- ①リズム機能
ADPCM機能
- ②4OP構造
SSG機能
- ③L R ステレオ出力
LFO機能

チップの性能としては、オペレータが4つ組み合わさっていますから音作りには十分なクオリティを持っているのですが、なにぶん FM 音が3声しか使えないのでこのチップだけでともに音楽をやろうとすると難があります。決定的に「弱い」部分は LFO 機能がチップになく、ピッチレートもトレモロも全部ソフトウェアで行わなければならない点です。

PC-8801のような16ビット機ならまだしも、PC-8800シリーズでこのような機能ま

でソフト側がサポートしなければならない
となると、CPUにかなりの負担を強いるこ
とになり、SRなどの拡張ベシックでは
6声部すべて音を出す演奏が「ヨタッて」
いました。

2 OPL(YM3526)または
OPL II(YM3812)

これはOPNとはまったく逆に、1ボイス当たりのオペレータの数を極限まで減らし2つにし、代わりに全体のボイス数を9ボイスに増やしたチップです。また、周波数固定ですが一応LFO機能も内蔵しています。さらに、通常のFM音源モードのほかのリズムモードというモードがあり、これを使うとFM音は6ボイスになります。が、余ったオシレータを使って同時に5つのリズム楽器音を出すことができます。

これを使ったシンセサイザボードには「MIDI MICRO MUSIC インターフェース」(ミュージック・ネットワーク)などがあります。これはPC-9800シリーズ向けのボードですが、OPLはチップの性能としては、音楽的に配慮されたクオリティの高いものですので、これについてはまた別の機会にレポートしたいと思います。

3 OPM (YM2151)

こちらは実際にシンセサイザの中で使われており、立派に1台の「楽器」として通用する性能を持ったチップです。

中身はOPNと同じく4オペレータタイプのFM音源ですが、発声部数が8ボイス。しかも、周波数、デプス、波形のコントロールができるLFO機能がつき、LRステレオ出力も合わせて、楽器を1台作るのに十分な性能を持っています。

これを使ったボード類はPC-8800シリーズ用に「響」(HAL研究所)というボードがあります。

4 MSX-AUDIO (YM8950)

こちらは MSX 用に作られた音楽チップで、FM 音源部と PCM サンプリグ部の 2 つのブロックに分かれています。

FM音源部は前述のOPL(YM3526)とまったく同じものです。それに加えてADPCMによるサンプリング音源部が内蔵されています。

図2 OPNAの内部ブロック

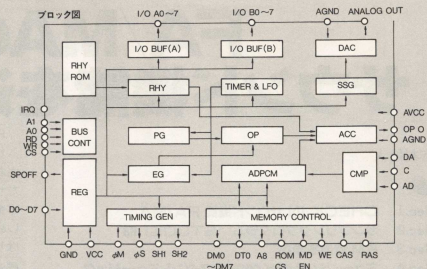


表 1 OPNA 主要機能

FM音源部	
発音数	6音
オペレータ	4コ
アルゴリズム	8コ
タイマー	A, B (YM2203に同じ)
LR制御	L, R, L+Rに設定可能
LFO機能	AM, PM, AMD, PMD, FREQ
その他	YM2203と同じ
SSG部	
発音数	3音
I/Oポート	あり
その他	YM2203と同じ
リズム部	
発音数	6音, PCMサンプリング
キーオン/ダンプ	イベント方式
LR制御	L, R, L+Rに設定可能
レベル制御	各楽器音およびトータルレベル
ADPCM	
サンプリングレート	1.6KHz〜16KHz
PCMビット数	8ビット
ADPCMビット数	4ビット
補間レート	55.5KHz
データメモリ	外付RAM, ROMおよびGPU側メモリ
外部メモリ	最大256Kバイト
LR制御	L, R, L+Rに設定可能

当然サンプリングには多くのメモリを消費します。MSX-AUDIOにはそのためのメモリコントローラの回路も内蔵されていますから、単体でサンプリング用に最大256KバイトのダイナミックRAMがパーソナルコンピュータ本体以外に使うことがで

此後

そしてOPNAについて

そしてOPNAの登場です。このICはこれまでのチップのよいところばかりを集め

図3-A OPNAのレジスタマップ

000H	SSG音源部	100H	ADPCM部
00FH	リズム部	10FH	
01FH	FM音源部 (共通部)		未使用
02FH	FM音源部 (ch1~ch3)		FM音源部 (ch4~ch6)
0B6H		1B6H	

て作られたチップです(図2,表1)。

まず,構造的には従来のOPNが2組内蔵され,それには不足していた機能やさらに発展した機能をつけ加えたものといえそうです。

OPNAの機能は大きく4つのブロック,

1. FM音源部
2. SSG 音源部
3. リズム部
4. ADPCM 部

に分けることができず(図3)。

以下,それぞれについて説明します。

1 FM音源部

FM音源部には新たにOPM相当のLFO機能とLRステレオ出力機能がつきました。以下はそれら新しいレジスタについての解説です。

新レジスタ \$2E, LFO

ここではLFOのON/OFFと発振周波数の設定を行います。レジスタの下3ビットがフリクエンシーコントロールのビットでその上のビット3がONフラグです。周波数は図4のように8段階設定できます。

\$B4~B6 PMS, AMS, LR

これは,各オペレータごとに設定できるLFOのためのパラメータです。PMSがピッチ・モジュレーション・デプス, AMSがアンプリチュード・モジュレーション・デプス, LRが出力チャンネルの左右切り替

図3-B SSG音源部

ADDRESS	MSB	LSB					意味
000H	Fine Tune					Channel A Tone Period	
001H	Coarse Tune						
002H	Fine Tune					Channel B Tone Period	
003H	Coarse Tune						
004H	Fine Tune					Channel C Tone Period	
005H	Coarse Tune						
006H	Period Control					Noise Period	
007H	/Noise		/Tone			/ENABLE	
008H	Mode		Level			Channel A Amplitude	
009H	Mode		Level			Channel B Amplitude	
00AH	Mode		Level			Channel C Amplitude	
00BH	Fine Tune					Envelop period	
00CH	Coarse Tune						
00DH	C		ATT	ALT	HLD	Envelop Shape Cycle	
00EH	I/O Port A					I/O port Date	
00FH	I/O Port B						

図3-C リズム部

ADDRESS	MSB	LSB	意味
010H	DM	リムショット	Dump/Rhythm Key ON
011H	タム	ハイハット	Rhythm Total Level
012H	シンバル	スネアドラム	LSIのTest Data
013H~017H	TEST		
018H	LR	Instrument Level	バスドラムのOutput Channel/Instrument Level
019H	LR	Instrument Level	スネアドラムのOutput Channel/Instrument Level
01AH	LR	Instrument Level	シンバルのOutput Channel/Instrument Level
01BH	LR	Instrument Level	ハイハットのOutput Channel/Instrument Level
01CH	LR	Instrument Level	タムのOutput Channel/Instrument Level
01DH	LR	Instrument Level	リムショットのOutput Channel/Instrument Level

01EH~01FHは未使用

図3-D FM音源部

FM音源部のアドレスは, ch1~ch6共通部を020H~02FHに, ch1~ch3とch4~ch6をそれぞれ030H~0B6Hと130H~1B6Hに割り当てています(レジスタの意味は同じです)。なお, アドレス020H, 120H~12FH は未使用です。

ADDRESS	MSB	LSB	意味
021H	TEST		LSIのTest Data
022H	LFO ON	FREQ.CONT	LFOのON/OFFとFREQ. Control
024H	TIMER-A		TIMER-Aの上位8ビット
025H	TIMER-A		TIMER-Aの下位2ビット
026H	TIMER-B		TIMER-BのData
027H	MODE	RESET B A	TIMER-A/BのControl ch.3のMode
028H	SLOT		Key ON/OFF CHはD0, D1, D2で指定
029H	SCH	IRQ ENABLE	IRQ イネーブル, SCH
02DH			プリスケールをセット
02EH			3分の1, 6分の1, 12分の1の選択
02FH			分周器を2分の1にセット

⇒図3-D FM音源部

030H(130H)	DT	MULTI	Detune/Multiple (33H/133H, 37H/137H, 3BH/13BHは未使用)
03EH(13EH)			
040H(140H)	TL		Total Level (43H/143H, 47H/147H, 4BH/14BHは未使用)
04EH(14EH)			
050H(150H)	KS	AR	Attack Rate/Key Scale (53H/153H, 57H/157H, 5BH/15BHは未使用)
05EH(15EH)			
060H(16EH)	AMON	DR	Decay Rate/AMON (63H/163H, 67H/167H, 6BH/16BHは未使用)
06EH(16EH)			
070H(170H)	SR		Sustain Rate (73H/173H, 77H/177H, 7BH/17BHは未使用)
07EH(17EH)			
080H(180H)	SL	RR	Sustain Level/Release Rate(83H/183H, 87H/ 187H, 8BH/18BHは未使用)
08EH(18EH)			

090H(190H)	SSG-EG	SSG Type Envelop Control (93H/193H, 97H/197H, 9BH/19BHは未使用)
09EH(19EH)		
0A0H(1A0H)	F-Num 1	F-Numbers/BLOCK
0A1H(1A1H)		
0A2H(1A2H)	BLOCK	F-Num 2
0A4H(1A4H)		
0A5H(1A5H)	F-Num 1 (ch.3)	ch.3-3SLOT, F-Numbers/BLOCK
0A6H(1A6H)		
0A8H(1A8H)	BLOCK (ch.3)	F-Num 2 (ch.3)
0A9H(1A9H)		
0AAH(1AAH)	FB	CONNECT
0ACH(1ACH)		
0ADH(1ADH)	L	R
0AEH(1AEH)		
0B0H(1B0H)	AMS	PMS
0B1H(1B1H)		
0B2H(1B2H)	PMS	AMS
0B4H(1B4H)		
0B5H(1B5H)	PMS	AMS
0B6H(1B6H)		

図3-E ADPCM部

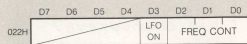
ADDRESS	MSB	LSB	意味
100H	START	REC	MEM DATA
101H	L	R	REPE AT
102H	START	ADR(L)	SP OFF
103H	START	ADR(H)	RESET
104H	STOP	ADR(L)	CONTROL 1
105H	STOP	ADR(H)	CONTROL 2
106H	PRESCAL	(L)	START ADDRESSの下位8ビット
107H	PRESCAL	(H)	STOP ADDRESSの下位8ビット
108H	ADPCM DATA		STOP ADDRESSの上位8ビット
109H	DELTA-N(L)		PRESCALEの下位8ビット
10AH	DELTA-N(H)		PRESCALEの上位3ビット
10BH	EG CTRL		ADPCMのバツファレジスタ
10CH	LIMIT	ADR(L)	DELTA-Nの下位8ビット
10DH	LIMIT	ADR(H)	DELTA-Nの上位8ビット
10EH	DAC DATA		Envelope Control
10FH	(PCM DATA)		Limit Addressの下位8ビット
110H	FLAG CONTROL		Limit Addressの上位8ビット
			DA変換時のデータレジスタ
			PCM-Dataの格納レジスタ(Read)
			各種フラグの制御

リードのみのレジスタ

ADDRESS	MSB	LSB	意味
xxH	FLAG		Status 0(A1="0"のとき) Status 1(A1="1"のとき)
FFH	ID NO		Status 2

図4 LFO レジスタ \$22 フリーケンシー

LFOの周波数とレジスタの関係



D2~D0	0	1	2	3	4	5	6	7
周波数(Hz)	3.98	5.56	6.02	6.37	6.88	9.63	48.1	72.2

図5 LFO レジスタ\$B4 デブス

PMS/AMS/LR: ADDRESS[0B4H~0B6H], [1B4H~1B6H]

D7	D6	D5	D4	D3	D2	D1	D0
0B4H~0B6H	L	R	AMS			PMS	
1B4H~1B6H							

PMS	0	1	2	3
変調度(セント)	0	±3.4	±6.7	±10
PMS	4	5	6	7
変調度(セント)	±14	±20	±40	±80

AMS	0	1	2	3
変調度(dB)	0	1.4	5.9	11.8

図6 LFO レジスタ \$60 AMON

Decay Rate: ADDRESS[060H~06EH], [160H~16EH]

Decay時のRate, 振幅変調スロットごとのON/OFFの設定を行います。

D7	D6	D5	D4	D3	D2	D1	D0
060H~06EH	AMON					DR	
160H~16EH							

DR: Decay Rateの設定を行います。

AMON: 振幅変調スロットごとのON/OFFの設定を行います。"1"のとき, ONとなります。

えです。図5のとおり、PMSは8段階、AMSは4段階のデプスが設定できます。
\$60~6E

このレジスタは、従来はエンベロープジェネレータのディケイトセットのためのものでしたが、ビット7に新しくLFOのうち、AMD(アンプリチュードモジュレーション)のON/OFFフラグが追加されました。

2 SSG

SSG部は従来のOPNとくに変更はありません。

3 ドラムセクション

リズム部は全部で6つの楽器、バスドラム、スネアドラム、シンバル、ハイハット、タム、リムショット、の音を出すことができます(図3-C)。使い方としてはOPLのリズム機能と変わらないのですが、実際のリズム音は、アナログのノイズジェネレータなどではなく、ちゃんとサンプリングした音が使え、またこれらリズム音は独自にボリューム設定、または出力チャンネルLRのセレクトができます。

リズムレジスタ \$10

このレジスタがリズム音のショットレジスタに当たります。ビット0から5までがそれぞれの楽器に対応しており、音を出す場合はそれら該当するビットに1を立てて書き込むと音を出すことができます。ビット7がダンブフラグといって、ここが0のときに該当する下のビット(0~5)が各音のONフラグになり、1になっている場合は該当するビットの音をダンブする(一時止める)働きをします。

リズムトータルレベル \$11

このレジスタはリズムセクションのボリュームをトータルにコントロールする働きを持つレジスタです。有効ビットはビット0から5までの6ビットで、減衰レベルの形で値を与えます。最大で-47.25dBのコントロールレンジを持ちます。

リズムレベル \$18~1D

各リズム楽器のボリュームとLRチャンネルの選択をこのレジスタで行います。ビット0から5までが各リズム音の個別のアッテネーターで、上ビット6から7までが出力チャンネルのLR切り替えスイッチに

図7 リズムレベルレジスタ \$10

	D7	D6	D5	D4	D3	D2	D1	D0
010H	DM		リム ショット	タム	ハイ ハット	シンバル	スネア ドラム	バス ドラム

バスドラム : バスドラムのリズム音を出します。

スネアドラム : スネアドラムのリズム音を出します。

シンバル : シンバルのリズム音を出します。

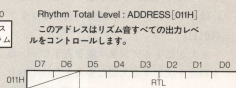
ハイハット : ハイハットのリズム音を出します。

タム : タムのリズム音を出します。

リムショット : リムショットのリズム音を出します。

DM : "1"のとき、ダンブ、"0"のとき、キーオンとなります。

図8 リズム、レベル、レジスタ \$11



RTL : リズム音すべてのレベルを6ビットで指定します。出力レベルの範囲は0~-47.25dBまでで0.75dBステップで制御します。

図9 リズムレベル \$18~1D

Output Select/Instrument Level : ADDRESS [018H~01DH]

このアドレスは、各リズム音の出力チャンネルと各インストルメントレベルを制御します。

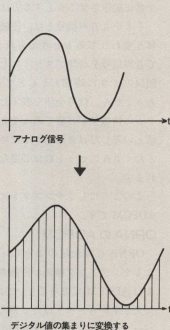
	D7	D6	D5	D4	D3	D2	D1	D0
018H~01DH	L	R			IL			

IL : 各インストルメントレベルを5ビットで表現します。各インストルメントレベルは、0~-23.25dBの範囲を0.75dBステップで制御します。

R : "1"のとき、Rich.に出力します。

L : "1"のとき、Rich.に出力します。

図10 PCMの原理



当たります。

つまりこれらのレジスタに値をセットすることにより、6つのリズム楽器音の音量のバランスと定位を自由に設定することができるわけです。

4 ADPCM

OPNAのほかのFM音源チップにない特徴にサンプリング機能が上げられます。OPNAには前述のMSX-AUDIOと同等のサンプリング機能が付加されています。

ADPCMとは

まず、ここではOPNAに採用されているサンプリングの方式「ADPCM」について説明したいと思います。

ADPCMとはAdaptive-Differential PCMの略で、通常知られているサンプリングの方式、PCMの変形版といった感じの方式です。

ご存じのとおり、普通PCMというのは音の波形を細かい時間で縦に分割して、各時間ごとの電圧をデジタル値で記憶しよう

とするものです (図10)。

分割する時間間隔がサンプリングレートで、一般に周波数で表します。これは大きいほどサンプリングの精度が高くなります (図11)。

また記憶するデジタル値を表すビット数も精度を決定する大切な要素となっており、12ビットから16ビットまでのデジタル化を普通のサンプリングキーボードなどでは採用しています (図12)。

サンプリングの精度を上げなければ、サンプリングレートを高くし、かつビット数を増やさなければならないわけですが、そうすれば結果として記憶させるためのメモリがたかくなることになります。

そこでメモリ量節約のために考えられたのがADPCMで、原理としては、波形の各時点での電圧値を表すのにその絶対値を用いず、前にサンプリングした値からの「差」を常に記憶させようとする方法です。

そもそも音声信号とは、連続した電圧の移り変わりであると考えられます。PCMで音声信号を時間でスライスした電圧値は、図14のように時間 t_1 と t_2 とで差はあまりありません。信号全体を表すために何ビットもの情報量が必要であっても「前の電圧値との差」だけを記憶するためにはそれほどたくさんのビット数は必要ないと考えられます。

このようにしてサンプリングする方式がADPCMです。

OPNAのADPCM

OPNAには上記のように、ADPCMによるサンプリング機能が搭載されています。またADPCMであっても、やはりサンプリングですらメモリはたくさん消費します。パーソナルコンピュータ本体のメモリだけでは十分な機能は発揮できません。そこでOPNAにはパーソナルコンピュータ本体とは別にメモリのコントロール回路を持ち、サンプリング専用メモリを使えるようになっていきます (図15)。

このOPNA内蔵のメモリコントローラは64Kまたは256KビットDRAMを直にコントロールでき、最大でDRAMを8個、合計256Kバイトがサンプリング専用に使えます。またこのメモリはサンプリング以外の目的でコンピュータ側からアクセスすることもでき、大容量のストレージメモリ

図11 PCMのサンプリングレート

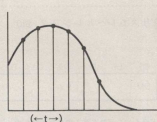


図12 PCMのビット

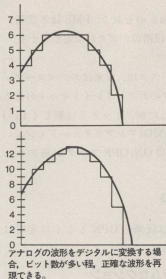
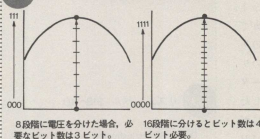


図13 サンプリングのメモリ量の関係



よりなめらかなサンプリングのためにはビット数を増やし、かつ、サンプリングの時間間隔を短くしなければならない。その分メモリも必要になる。

図14 ADPCMでのサンプリング

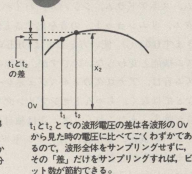
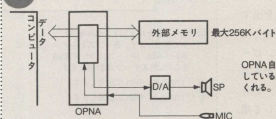


図15 OPNAのADPCM方式



OPNA自身がコンピュータとは別に独自にメモリを管理している。サンプリング/再生はすべてOPNAがやってくれる。コンピュータはコマンドを出すだけでよい。

図16 レジスタ \$00

コントロールレジスタ1: ADDRESS[100H]
このレジスタは、ADPCM音声分析/合成の起動、サウンドメモリとのアクセスを制御します。

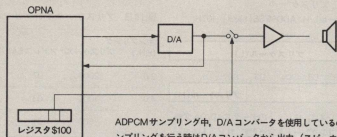
D7	D6	D5	D4	D3	D2	D1	D0
100H	START	REC	MEM DATA	REPEAT	SP OFF	RESET	

- RESET : サウンドメモリをデータ源として、ADPCM音声合成を実行中に「1」とすると初期状態に戻ります。ただし、REPEATは「0」とします。
- SPOFF : 「1」のとき、SPOFFは「1」となり、ADPCM分析、AD変換時のスピーカへの信号をオフとします。
- REPEAT : 「1」のとき、サウンドメモリの同一アドレス区画をくりかえしてアクセスして、ADPCM音声合成を行います。
- MEM DATA : 「1」のとき、サウンドメモリにアクセスが可能となります。
- REC : 「1」のとき、ADPCM音声分析を行い、サウンドメモリにデータの書き込みを可能にします。

コントロールレジスタ2: ADDRESS[101H]

D7	D6	D5	D4	D3	D2	D1	D0
101H	L	R		SAMPLE	DATA	RAM TYPE	ROM

図17 OPNAのSPOFF回路ブロック図



ADPCMサンプリング中、D/Aコンバータを使用しているため、サンプリングを行う時はD/Aコンバータから出力（スピーカetc）される信号をカットする。そのフラグがSPOFF。

図18 スタート/ストップアドレス

スタートアドレスL/H: ADDRESS [102H], [103H]

下の図で、BANKとは8個のDRAMのチップセレクトを意味します。

・DRAM (256Kバイト)
コントロールレジスタ2のRAM TYPEで"0"を設定したとき、スタートアドレスL/Hは次のようになります。

D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	0 0 0 0 0
START ADDRESS (H)	START ADDRESS (L)	
2 ⁸ 2 ⁷ 2 ⁶ A8 A7 A6 A5 A4 A3 A2 A1 A0	A8 A7 A6 A5 A4 A3 A2 A1 A0	
BANK	CAS ADDRESS	RAS ADDRESS

・ROM, DRAM (256Kバイト)
コントロールレジスタ2のRAM TYPEで"1"を設定したとき、スタートアドレスL/Hは次のよ

うになります（*のついたデータビットは、ストップアドレス（H）と同じ値にしてください）。

D7*D6*D5*D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	0 0 0 0 0
START ADDRESS (H)	START ADDRESS (L)	
2 ⁸ 2 ⁷ 2 ⁶ A8 A7 A6 A5 A4 A3 A2 A1 A0	A8 A7 A6 A5 A4 A3 A2 A1 A0	
BANK	CAS ADDRESS	RAS ADDRESS

ストップアドレスL/H: ADDRESS [104H], [105H]

下の図で、BANKとは8個のDRAMのチップセレクトを意味します。

・DRAM (256Kバイト)
コントロールレジスタ2のRAM TYPEで"0"を設定したとき、ストップアドレスL/Hは次のようになります。

D7 D6 D5 D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	1 1 1 1 1
STOP ADDRESS (H)	STOP ADDRESS (L)	
2 ⁸ 2 ⁷ 2 ⁶ A8 A7 A6 A5 A4 A3 A2 A1 A0	A8 A7 A6 A5 A4 A3 A2 A1 A0	
BANK	CAS ADDRESS	RAS ADDRESS

・ROM, DRAM (256Kバイト)
コントロールレジスタ2のRAM TYPEで"1"を設定したとき、ストップアドレスL/Hは次のようになります。

D7*D6*D5*D4 D3 D2 D1 D0	D7 D6 D5 D4 D3 D2 D1 D0	1 1 1 1 1
STOP ADDRESS (H)	STOP ADDRESS (L)	
2 ⁸ 2 ⁷ 2 ⁶ A8 A7 A6 A5 A4 A3 A2 A1 A0	A8 A7 A6 A5 A4 A3 A2 A1 A0	
BANK	CAS ADDRESS	RAS ADDRESS

などとして使うこともできます。

ADPCM関係のレジスタとその機能

コントロールレジスタ1 \$00

アドレス\$00はコントロールレジスタで、ADPCMによる音声分析（サンプリング）、合成、外部メモリへのアクセスなどの機能をコントロールします。

ビット0: RESET

外部メモリをデータ源としてADPCM音声合成中に「1」を書き込むと合成処理が初期化されます。このときビット4のREPEATは0にしておきます。

ビット3: SPOFF

OPNAチップには外部出力端子としてSPOFF（スピーカ・オフ）という端子があります。このビットはその端子に直結しており、音声のサンプリング中に外づけD/Aコンバータからの出力が外部に洩れないような回路を付加するために使われます（図17）。

ビット4: REPEAT

ここに「1」を立てることにより外部メモリの同一アドレス区間を繰り返してアクセスし続けます。音声合成（再生）にループを持たせることがこれでできます。

ビット5: MEM DATA

ここが「1」のときOPNAから外部メモリへのアクセスが可能になります。

ビット6: REC

ここが「1」のときOPNAから外部メモリへの書き込みが可能になります。

ビット7: START

「1」のときADPCMの合成/サンプリングを始めます。

コントロールレジスタ2 \$01

ビット0: ROM

このビットは外部メモリの選択を行います。1のとき外部メモリとしてROMをアクセスし、0のときDRAMをアクセスします。

ビット1: RAM-TYPE

DRAMのタイプを指定します。

ビット2: DA/AD

このビットが「1」のときDAC-DATレ

ジスタ (後述) のデータを外づけ D/A コンバータから出力します。また、このビットが 0 かつ SAMPLE ビットが 1 のとき PCM による AD 変換 (サンプリング) を行います。

ビット 3 : SAMPLE

このビットを 1 にした時点から AD 変換 (サンプリング) が始まります。

ビット 6, 7 : L/R

D/A 変換された信号 (サンプリング信号) の出力チャンネルを指定します。

レジスタ \$02~\$05

スタート/ストップアドレス (図18)

サンプリング時に外部メモリの何処から何処までをサンプリング/合成に使用するかを指定します。アドレスの下位 5 ビットは常にスタートについては 0 を、ストップについては 1 を書き込みます。

レジスタ \$06~\$07

プリスケール (図19)

AD, DA 変換時のサンプリングレートを決定します。

レジスタ \$08

ADPCM-DATA (図20)

パーソナルコンピュータ側が独自に OPN A に接続された D/A コンバータを用いて DA/AD 変換を行う場合と、外部メモリに直接アクセスする場合にこのレジスタを介してデータのやりとりを行います。

レジスタ \$09~\$0A

DELTA-N (図21)

ADPCM での音声合成時に、各サンプリング間を補間するための係数を与えます。この値はサンプリング時と音声合成時とで異なるビットで再生を行うときに使用します。

レジスタ \$0B

ENVELOP (図22)

ADPCM 音声合成出力のレベルコントロールを行います。

レジスタ \$0C~\$0D

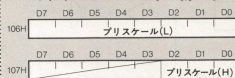
リミットアドレス (図23)

リミットアドレスへ到達したら 0 番地へ

図19

図19A プリスケール

プリスケール L/H : ADDRESS [106H], [107H]



このアドレスは、11 ビットのデータで AD/DA 変換時のサンプリングレートを定めます。11 ビットのデータのうち、下位をアドレス [106H]、上位をアドレス [107H] で指定し、プリスケールの値は下のような構造になります。

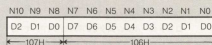


図19B プリスケールサンプリング対応表

サンプリング KHz	プリスケール	アドレス \$07	アドレス \$06
2.0	2000	07	D0
4.0	1000	03	E8
8.0	500	01	F4
16.0	250	00	FA

図20

ADPCM DATA

ADPCM DATA : ADDRESS [108H]

このアドレスは、ADPCM 分析/合成をプロセスが行うとき、または、サウンドメモリをアクセスするときのバッファレジスタです。

このアドレスは、他のアドレスと異なり、READが可能です。

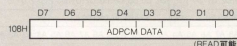
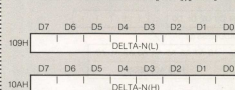


図21

図21A DELTA-N

DELTA-N L/H : ADDRESS [109H], [10AH]



このアドレスは、ADPCM 音声合成時に各サンプリングレート間を補間するための係数を制御します。データビットは 16 ビットで、下位 8 ビットをアドレス 109H、上位 8 ビットをアドレス 10AH で指定します。

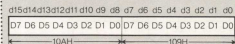


図21B DELTA-N対応表

サンプリング KHz	Delta -N	アドレス \$0A	アドレス \$09
2.0	2359	09	37
4.0	4719	12	6F
8.0	9437	24	DD
16.0	18874	49	BA

図22 ENVELOP CONTROL

ENVELOP CONTROL: ADDRESS[10BH]

このアドレスは、ADPCM音声合成の出力レベルを制御します。

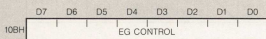


図24 DAC-DATA

DAC-DATA: ADDRESS[10EH]

このアドレスは、DA変換を行うためのデータレジスタです。

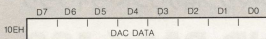
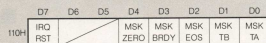


図26 FLAG-CONTROL

FLAG-CONTROL: ADDRESS[110H]



MSK TA: "1"のとき、タイマーAの動作に関係なくタイマーAのフラグを"0"にします。

MSK TB: "1"のとき、タイマーBの動作に関係なくタイマーBのフラグを"0"にします。

MSK EOS: "1"のとき、ADPCM音声分析/合成、サウンドメモリのREAD/WRITEの終了、AD変換時の終了を指定するフラグを"0"にします。

MSK BRDY: "1"のとき、ADPCM音声分析/合成、サウンドメモリのアクセス時のデータの書き込み要求、読み出し要求フラグを"0"にします。

MSK ZERO: "1"のとき、ADPCM音声分析中の無音状態を示すフラグを"0"にします。

IRQ RST: "1"のとき、書き込むと、D4~D0は無視されるが、すべてのフラグは"0"になります。

図23 リミットアドレス

リミットアドレス L/H: ADDRESS[10CH],[10DH]

リミットアドレスに到達したら、0番地にもどります。

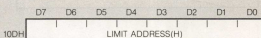
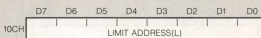
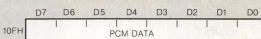


図25 PCM-DATA

PCM-DATA: ADDRESS[10FH]

このアドレスは、AD変換時に変換済みデータを格納するためのレジスタです。このアドレスはREADのみで、WRITEはできません。



もどります。

レジスタ \$0E

DAC-DATA (図24)

DA変換を行うときのデータレジスタです。

レジスタ \$0F

PCM-DATA (図25)

AD変換された変換済みデータを格納するレジスタです。読み出し専用で書き込みはできません。

レジスタ \$10

FLAG-CONTROL (図26)

各種フラグのコントロールで、初期設定

時ビット4, 3, 2は「1」、その他は「0」となります。

ビット0: タイマーAマスク

ここが「1」のとき、タイマーAの動作に関係なくタイマーAのフラグを「0」にします。

ビット1: タイマーBマスク

ビット0と同じく、ここが「1」のとき、タイマーBの動作に関係なくタイマーBのフラグを「0」にします。

ビット2: EOSマスク

ここが「1」のときEOSフラグをマスクして「0」にします。EOSフラグはADPCMサンプリング/合成の終了、外部メモリへのリード/ライト終了、AD変換処理終了などを表すフラグです。

ビット3: BRDYマスク

「1」のときADPCMサンプリング/合成、外部メモリのアクセス要求フラグを「0」にします。

ビット4: ZEROマスク

「1」でADPCM音声分析、無音フラグを「0」にします。

ビット7: RESET

このビットを「1」にしてデータを書き込むとビット0から4までは無視され、また、すべてのステータスフラグを「0」にします。

ステータスレジスタ

ステータスレジスタは2つあり、それぞれアドレスレジスタを読み出すことにより

図27 A ステータスレジスタ 0

STATUS-0(READ)							
D7	D6	D5	D4	D3	D2	D1	D0
BUSY						FLAG B	FLAG A

図27 B ステータスレジスタ 1

STATUS-1(READ)							
D7	D6	D5	D4	D3	D2	D1	D0
BUSY		PCM BUSY	ZERO	BRDY	EOS	FLAG B	FLAG A

読み出せます (図27)

ステータス-1(A0=0)

ビット0~1: タイマーフラグ

タイマーA、Bのオーバーフローフラグです。

ビット7: BUSY

OPNAがデータを内部レジスタへロードする最中「1」になります。

ステータス-2(A0=1)

ビット0~1: タイマーフラグ

ステータス0と同じく、もう組のタイマーのためのフラグです。

ビット2: EOS

ADPCM サンプリング/合成が終了したとき、AD/DA 変換時にサンプリングタイムが経過したとき「1」になります。

ビット3: BRDY

ADPCM サンプリング/合成時、2データ分の処理が終了したとき、また、外部メモライト/リードで1データ分の処理が終了したときに「1」になります。

ビット4: ZERO

ADPCM サンプリング中、サンプリング信号が無音の状態のとき「1」になります。

ビット5: PCM-BUSY

ADPCM サンプリング/合成中「1」になります。

ビット7: BUSY

データをレジスタにロード中「1」になります (ステータス-1に同じ)。

以上がFA/MAのサウンド機能の概要です。前述したサンプリング機能に関連してオーディオラボから発売されたユニークなサンプリング用CDのご紹介を最後に付け加えておきましょう。

サンプリング・マシン用サウンド・ライブラリ

SOUND PACK 128

オーディオラボ ASP-9801 9,800円

「ワンワン、ニヤンニヤン、ジャジャジャジャーン」とサンプリングが流行していて、パソコンやサンプリング・シンセサイザーでサンプリングを楽しんでいる方も多いと思います。そんなあなたにピッタリなのが、このサンプリング用CD [SOUND PACK 128] です。

このCDには、ピアノやトランペット、バイオリン、ギター、ドラム、オーケストラの音など色々な楽器の音が128バリエーション収録されています。各音色が2オクターブほど収録されているので、組み合わせることにより、1パートのメロディなら簡単にできてしまいます。

直接、音を鳴らしてマイクで拾うサンプリング方法がありますが、このサンプリングは音の瞬間的な問題もありタイミングがとても難しいものです。まして楽器などの音を組み合わせるには、ノリやズレなどで常に同じ状況とは限りません。また、余分な音も入ってしまうことも考えられます。でも、CDプレイヤーのラインアウトから

らサンプリングすれば、タイミングがずれることなくノイズの問題からも解放され、常に一定の条件のサンプリングができます。想像しての音作りサンプリングするのではなく、音を聴き、好みの音を使用目的に応じて選べます。

たとえば、バイオリン、チェロの音を幾つも重ねて組み合わせれば、それで、もうオーケストラ演奏がパソコンから流れてくるのです。タムやスネアドラムの音の組み合わせでは、ドラム・パターンも作れます。ピアノの音の組み合わせをサンプリングすれば、バッハやベートベン、シューベルトなどの名曲も楽しめるというわけです。

オーケストラ・ヒットやストリングス・ヒットなども、本格的に収録されていますので、十分でしょう。

どの音をどのように使うかということですが、いくらでも楽しめるので、これさえあれば、あなたも名コンダクターになれます。

Sec.2

サンプリングプログラム STUDIO 88

これはOPNAのADPCM機能を使ったサンプリングプログラムです。PC-8801/FA/MA付属の拡張ベーシックには、ADPCM機能をコントロールするための新ステートメントが用意されています。サンプリングはのうちPCM、PCM PLAY、PCM RECORD、PCM LOAD、PCM SAVEなどのステートメントで簡単なPCMレコーディング、再生を行っています。

プログラムの機能と概要

1 PCM

ADPCM関係の初期設定を行うためのステートメントです。パラメータとして4つの数値を設定します。それぞれ音量、定位、サンプリングレート、実行モードと呼ばれます。

音量：0から255までの値で、ADPCM再生時の出力レベルを設定します。

定位：1から3までの値を取り、ADPCM再生時のLR出力チャンネルを設定します。1-R(右)、2-L(左)、3-L+R(左右)に出力されます。

サンプリングレート：サンプリングレートを指定します。指定する値は4-(4KHz)、8-(8KHz)、16-(16KHz)の3種類です。

実行モード：ほかのBASICステートメントに対してADPCMサンプリング、再生をバックグラウンドで行うかどうかを指定します。1でバックグラウンドモード、0でフ

ォアグラウンドモードになります。

2 PCM RECORD

ADPCMによるサンプリング(レコーディング)を行います。設定されるパラメータは録音開始時刻、録音時間の2種類です。

このパラメータ上の「時刻」は0.1秒単位で指定され、「開始時刻」はサウンドメモリ(サンプリング用にOPNAに外づけされたD-RAM)の先頭アドレスを0秒として何秒間過ぎたところにサンプリング開始点を持ってくるかを指定します。たとえばサウンドメモリの頭から10秒間をレコーディングさせる場合「0, 100」を設定します。

3 PCM PLAY

PCM RECORDによってサンプリングされたデータを再生します。設定パラメータはRECORDと同じく2つでそれぞれ再生開始時刻、再生時間です。

4 PCM SAVE

OPNAのサウンドメモリのデータをディスクへセーブします。パラメータはファイル名、開始時刻、終了時間の3つです。開始時刻、終了時間の意味はPCM RECORDに同じです。

5 PCM LOAD

PCM SAVEによってセーブされたデータを再びOPNAのサウンドメモリへロードしてきます。パラメータは、ファイル名、開始時刻の2つで、それぞれの意味はPCM RECORDに同じです。

```

1000 '-----
1010 '
1020 '   ADPCM Record & Play Back
1030 '
1040 '-----
1050 WIDTH 80,25
1060 PRINT
1070 PRINT"  1.   Record
1080 PRINT"  2.   Play
1090 PRINT"  3.   Save
1100 PRINT"  4.   load
1110 PRINT
1120 INPUT " select No. ",X
1130 IF X<1 OR X>4 GOTO 1060
1140 '
1150 ON X GOSUB *REC,*PLAY,*SAV,*LOD
1160 GOTO 1050
1170 '-----
1180 '   Record
1190 '-----
1200 *REC
1210 PRINT:PRINT
1220 INPUT " hit [ RETURN ] ".X$
1230 PRINT " Now Recording .... "
1240 PCM 255,3,16,0
1250 PCM RECORD 0,16
1260 PRINT:" Rec End "
1270 GOSUB *PLAY
1280 RETURN
1290 '-----
1300 '   Play back
1310 '-----
1320 *PLAY
1330 PRINT:PRINT " Now Playing "
1340 PCM 255,3,16
1350 PCM PLAY 0,160
1360 RETURN
1370 '-----
1380 '   Save
1390 '-----
1400 *SAV
1410 FILES
1420 PRINT
1430 INPUT "Input SAVE filename ",X$
1440 IF X$="" THEN RETURN
1450 PCM SAVE X$,0,160
1460 RETURN
1470 '-----
1480 '   Load
1490 '-----
1500 *LOD
1510 FILES
1520 PRINT
1530 INPUT "Input LOAD filename ",X$
1540 IF X$="" THEN RETURN
1550 PCM LOAD X$,0
1560 RETURN

```


Sec.3

FM音源を学ぶ

音とは!?

「このパソコンは、アラ不思議。これだけでわたしも作曲家」とCMで歌われているとおり、PC-8801FA/MAがサウンド機能に優れていることはすでに前の項で説明しました。しかし、「FM 6音+リズム6音」ということにピンときている人は少ないはず。この「FM音源」という言葉、耳にしたことはあるけれど、一体何なのか？ あるいは、ラジオ放送にもFMっていうのがあるなあ？ 確か、シンセサイザなどで使っているハズなのだけど、パソコンからシンセサイザと同じ音が出るのかなあ？ ともろもろの疑問を持っている方も多いことでしょう。

そこで、今回FA/MAをより知る上で、「FM音源」についてやさしく解説してみたいと思います。まずは、「音とは何であるのか？」という説明からしていくことにしましょう。

音のしくみ

音は、空気の振動によって起こります。その振動が空気中を伝わって耳の鼓膜に到着した時点で、音として認識されます。アコースティック・ギターの場合で説明をしてみましょう。弦をはじくと、弦が揺れて、空気を振動させます。

その際に関係するのが、

①どの位の強さで弦をはじいたか

②弦は太いか細いか

③弦を押さえている場所によって、どれくらい長い弦が振動したか

ということですが、これが音の3つの要

素である、[大きさ・音色・高さ]なので、図1では、波が0からのぼっていき、上まで達したら今度は下にさがっていきます。これを1周期といいます。この波の回数が1秒間に何回あるかで音の高さが決まります。このことを周波数と呼び、Hz(ヘルツ)という単位で表します。図1では波が1秒間に1回なので1Hz、図2では波が3回なので3Hzというわけです。

周波数は、音(程)が高いほどその値が大きくなります。たとえば、ピアノの中央付近の「ラ」の音は440Hzで、さらに1オクターブ高い「ラ」の音は880Hzになります。

図3には2つの波が書かれています。波の一番高いところ(山)と一番低いところ(谷)の差を振幅といいます。振幅は音の大きさを表し、振幅が大きいほうが音が大きくなります。この場合、A図のほうがB図より波の高さの差が大きいため、振幅が大きく、つまり、音が大きいということ。音の大きさには、ワットやホン、デシベル(dB)といった単位が使われています。

音の3要素の中で音の高さは周波数で表され、大きさは振幅で、そして音のボイントともいえる音色はその波形によって決まります。波形はアナログ・シンセサイザなどによく出てくるもので、[正弦波・三角波・矩形波] (図4)の3つに大別することができます。波形は、波の形(ウェーブ・フォーム)が丸いほど柔らかい音になり、尖っているほど鋭い音が出ます。本来、最も純粋であり、すべての音の基本となるのが、ウェーブ・フォームが丸い正弦波(サイン波)です。信じられないことに、鋭い音もこの柔らかい波形から作られるのです。

図5では2つの違う正弦波があります。元となるAの波形に、Aの2倍の周波数で

2分の1(半分)の音量のBの波形を足すと(合成する)、Cのような波形ができます。この波形は、かなり三角波に近くなっています。この波形にさらに、Aに対して周波数が3倍4倍5倍の波形を加えていけば、どんどんと三角波に近づきます。このようなAの元の波形を基本波といい、Bのような基本波に対して、周波数の倍数のものを倍音(ハーモニクス、高調波ともいいます)と呼んでいます。音は倍音のいくつかの組み合わせにより色々な音が作り出せるということです。

FM音源とは

近ごろ、さかんにシンセサイザや電子ピアノなどの電子楽器などで、FM音源やSA音源、PCM音源と色々な音源が搭載されているとカタログに書かれていますが、その音源は、どのような違いがあるのでしょうか？ また、どうして色々な方式の音源があるのでしょうか？

これらの各音源は、音を合成するしくみがさまざまにできており、一長一短があります。そのため、メーカーは各機種のコンプライアントに合った最適な音源方式を選んで使っています。

PC-8801FA/MAには、デジタル音声合成方式として最もポピュラーであるFM音源を搭載しています。FM音源を搭載しているシンセサイザの代表選手として、ヤマハのデジタル・シンセサイザ「DX-7」シリーズがあります。しかしPC-8801FA/MAは、その「DX-7」とまったく同じ音が出るというわけではありません！ 同じ音源方式を使っている。つまり音を作り出すしくみが同じであるというわけですね。

FM音源のFMというのは、FREQUENCY MODULATIONの頭文字のことで、周波数変調ということを表しています。これは簡単に説明すると、元の音に、ある別の音を影響、作用させて(変調)、元の音に変化をつけることです。たとえば、合唱団でアルトの人が歌っているとしします。そこに別の人が来て、そのアルトの人をくすぐったとします。

するとそのアルトの人の声は笑い声に変わり、音が高くなりソプラノの声になります。これが周波数変調の簡単な原理です。図7はそれを表しています。また、図8では2人にくすぐられていますので、よりカン高い笑い声になっているというわけです。下の図は、絵のそれぞれの関係を表しているもので、アルゴリズムといえます。DX-7のパネルなどに書かれていて、FM音源の仕組みの基本となるものです。

FM音源ではキャリア(変調をうける信号)を音源にして、モジュレータ(変調する信号)によって音を作ります。ですから、くすぐられるアルトの人は元になるキャリアであり、くすぐる人はキャリアを変化させるモジュレータということです。

FM音源では、さきほどのアルトの人やくすぐる人のことをオペレータと呼んでいます。4オペレータなどとカタログに書かれているのは、1つの音色に対して、オペレータが4つあるよという意味です。そのオペレータはそれぞれ、最もシンプルな波形のサイン波(正弦波)が基本になっています。しかし、そのサイン波が複数に変調しあうことにより、色々な複雑な音を出すことが可能になります。ガラスの割れる音や、紙の破れるような複雑な音や金属系の音色は、FM音源が得意とする音色です。

オペレータの波形は、サイン波と説明しましたが、正確にはサイン波が1波分だけROMに記憶されていて、それを読み出す時間を変化させることによって色々な波形にすることも可能です。また、この読み出しのスピードにより、出てくるサイン波の周波数(ピッチ)が変わり音程を決めています。そして、そのスピードをコントロールするのがPG(Phase Generator)と呼ばれるものです。キャリアもモジュレータもそれぞれ別のまったく違った動きをしていますが、同じものなのです。同じサイン

図1 周波数のしきみ

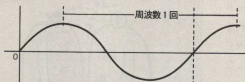


図2

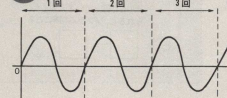


図3

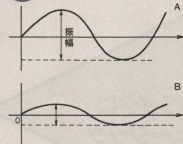
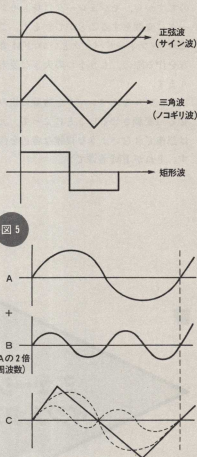


図4 音の要素とそのコントロール

音の要素	コントロールするもの	シンセサイザ
高さ	周波数	VCO (ホルテージ コントロールド オシレータ)
音色	波形	VCF (ホルテージ コントロールド フィルター)
大きさ	振幅	VCA (ホルテージ コントロールド アンプリファイヤー)
鳴り方 消え方	音量の時間的変化	EG (エンベロープ ジェネレータ)

図5 音の波形・3種類



三角波は基本波Aと倍音Bの和によってできる

波でも違う働きをして複雑に影響(変調)しあうことによって、ノコギリ波や矩形波などが得られるのです。この影響力(変調入力)のレベルによってさまざまな音を出し、変調のレベルが上がると、波の歪み(ひずみ)も大きく変化するので、より複雑な音となり、さらにレベルが上がると元のサイン波からは想像できないようなまったく違

った音になり、ノイズに近いような音にもなります。

FM音源では、PGが常に一定の間隔でサイン波を読み出しているため、周期性が一定に存在し、正確な音程を保っています。アナログ・シンセサイザなどによく起きる、音程の不安定さがないわけです。また、モジュレータからキャリアと2種類のオペレ

ータが必要なわけではありません。モジュレータもキャリアも、元はといえば同じものですから、モジュレータ自身もモジュレータに変調をすることができます。これをセルフ・フィードバックといい、FM音源で音を作る際の、もう1つの大きな要素となります。

シンプルなオペレータを複雑に組み合わせる(変調させる)ことによって、元からは想像できないような複雑な音色を作り出す。それがFM音源です。

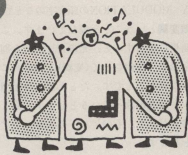
図7



アルトで歌っている人がすぐ
られると、ソプラノの声になる



図8



2人にすぐられる
とよりカン高いソ
プラノになる



Sec.4

ミュージシャンライクに インスタントミュージック

誰にでも、 すぐ音楽が楽しめる!

NECからFA/MAと同時に出されたミュージックソフト「Instant Music」は、よくいったもので、名前のとおりすぐに音楽ができてしまいます。

音楽とは、読んで字のごとしで、「音を楽しむ」ものです。誰にでも、このソフトなら簡単に音楽を楽しめます。

コンピュータのキーボード、マウスのどちらかで操作が選択できるようになっています。音符もないし、リズムもマウスで幾つかのパターンで選べるし、音色さえもマウスで選ぶことができます。そして、音量、テンポなども、画面上でマウスで操作できるので、演奏中でも一目瞭然にわかります。ですから、この「Instant Music」があれば、コンピュータについての知識も音楽の

知識もまったくいらずに、自分の感性がそのまま表現でき、音楽が楽しめるというわけです。

新しいコンピュータ・ ミュージック

「Instant Music」は、PC-8801FA/MAのパワーアップした内蔵音源をフルに利用して楽しむことができます。また、今までのPC-8800シリーズを持っている人には、サウンド・ボードが用意されていますので、これを搭載するとFA/MAに遜色ない演奏が楽しめます。今までのMIDIを使用したコンピュータ・ミュージックソフトに比べたら、ソフトウェアだけで音楽が気軽に楽しめます。

PC-8801FA/MAにはFM音が6つ、SSG音が3つ、リズム音が6つの性能を持っているので、十分にサウンドに厚みが出

ます。ですから、FA/MAユーザーの方はぜひ、「Instant Music」を楽しんでいただきたいと思います。

では、実際に簡単に操作できるマウスを使ってみるの紹介をしてみましょう!

電源を入れるだけで 楽しめる

電源を入れ、ディスクにフロッピーを差し込みリセット・ボタンを押すと立ち上がり「Instant Music」の画面が表示され、しばらくするとテーマソングの演奏が始まります。機能のほとんどの操作がカーソルで行えます。このカーソルの移動をキーボードのキーにするか、マウスにするかを最初に決めます。

その際もESCキーで表示され、メニューバーから選択すると簡単に表示されます。マウスには、左と右に2つのボタンがあり

ます。この2つのボタンをメニューと選択すると同時に持っていくカーソルを動かして、クリックして操作の選択を行います。

まずはテーマソング

最初に演奏されるテーマソングは2小節ほどの簡単なメロディがリズム1パート、ベースやほかの楽器で4パート、合計4パートで繰り返して演奏されます。ここでの画面を見て、「エッ！これは何を表しているのかな？」と思われるはずです。画面上に赤、緑、青、黄とカラフルなカラーのブロックのようなものがちらちら画面上に散らばっている、としか思えないのですが、これが「Instant Music」での音符。そうです、この画面が「Instant Music」の世界での楽譜なのです。

これなら、誰もが簡単に書くことができます。また「Instant Music」は、コンピュータが自動的に判断して変な音やはずれた音を演奏しないように編集してくれます。これが一番の特徴といえます。ですから、人工知能内蔵コンピュータ・ミュージックソフトとってしまっても過言ではありません。

まずは、音を止めてみましょう。画面の右上のスタート/ストップスイッチにカーソルをおき、マウスの左ボタンをクリックすると演奏がとまり、もう一度クリックすると演奏が再開されます。曲の途中から演奏されることがないので、途中から始まり、何がなんだかわからなくなることはありません。画面の下には、各楽器名が書かれ、その下に長く棒が引く張ってあります。これが各楽器ごとのボリュームです。ミキサー機能が簡単に操作できます。音符を表しているブロックと色が統一されているので、どのパートの音かが、すぐにわかります。バーの色が占める面積が長いほど音が大きくなります。カーソルを動かして、クリックした位置で音量が決まります。1パートずつ聴いてみたいときはほかのバーを小さくしてボリュームを下げれば、ほかのパートは鳴りません。ボリューム・バーの右に三角が3つありますが、これはオクターブの切り替えて、□がその楽器の標準の音域、△だと1オクターブ上、▽だと1オクターブ下になります。2オクターブの差が

あります。各楽器でも2オクターブでは、随分、音のイメージが変わりますので、色々な音色を試すことができます。

つぎはテンポです。画面の左上に赤いバーがあります。これがテンポバーです。赤い部分が多いほどテンポが速くなります。

そのテンポの動きも画面の下の部分で赤い線が動き、目でも確認できます。演奏を中止せずにテンポを次々に変えることができるので、イメージはすぐにつかめます。

音色を変えて楽しもう

「曲は、アレンジしだいたい」とよくいわれています。バックギンやベース、ドラムの入れ方だけではなく、演奏する楽器により同じメロディでも随分と曲のイメージが変わるものです。そこで楽器を変えてみることにしましょう。音色を変えたいトラック・コントロール・ボックスにカーソルをもっていくをクリックします。そして、画面の一番上のメニューバーにいきますと、写真2のように上にメニューが表示されます。そこ「サウンド」でまた、クリックすると写真3の画面が表示されます。ここでは、色々な音色が入っています。ピアノやトランペット、Eベースなどの楽器だけではなく、琴やベルの音も入っていて、十分に音色選べます。また、楽器音以外のUFO、虫、レーザー音などもあります。この曲作りだけではなく、効果音を使い色々な音を出すという楽しみ方もあります。この中からよい音色を選びカーソルを動かして、クリックすれば、もう音色指定がされています。このようにして、音色を変え、テンポを変え、音程をオクターブずつ変えれば、かなり最初のテーマソングから変わります。でも、まだまだ、それだけでは音楽の楽しみのホンの少しにしかなりません！次はリズムです。

カーソルをメニューのオプションにもっていき写真4の画面が表示されます。ここで曲の長さ、拍子、リズムパターン、リズムの音量を決めます。リズムの右の黒くリズム名が書かれているところにカーソルをもっていく、クリックをしてみると、リズム名が変わります。次々にクリックすると、色々なリズム名が出てきます。ここで、リズムを「ロック2」にしてみました

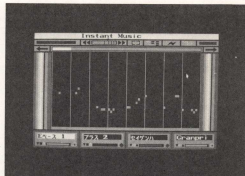


写真1 「Instant Music」演奏画面

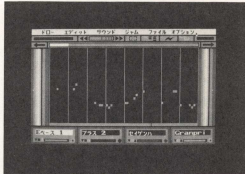


写真2 メニュー表示画面



写真3 音色表示画面

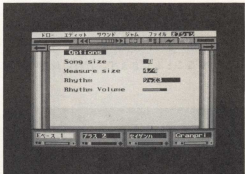


写真4 リズムパターン表示画面

う。そしてテンポを少し速めにし、リズムの音量を大きめにしていれば、最初のテーマソングも、もう立派なハードロックになってしまいました。

実際に音符を入力する

さて、いよいよ音符の入力です。ここから

らがホントの自分の音楽を楽しめる世界になるわけです。

メニューのファイルを選べば、写真5の画面が表示されます。ここで、ロード、セーブの作業を行います。ロードのメニューを選べばファイル名が表示されます。「Instant Music」にはあらかじめ、色々な曲が数曲入っています。その中から自分の頭にある曲のメロディ、リズムなどのイメージに近い曲を選び、それを少しずつ変えて音符を入力するという方法もとれます。直接は何もないところに入力する方法もありますが、慣れるまでは元の曲を少しずつ変えていく方法が簡単に親しみやすいでしょう。

音符を入力する楽器を選び、カーソルを動かします。クリックして設定したら、カーソルを音符ウィンドウ内にもっていきます。そこで位置を決めてクリックすると、そのクリックした位置に音符のブロックが移動します。このようにして1音ずつ移動していきますが、各楽器ごとの設定なのでパートが途中で崩れることもありません。また、コントロール・スイッチの4番目のところにます目が1個、2個、3個と書かれています。そこで選択すれば、1音ずつだけではなく、一度に2音、3音と入力していけるので、手間がかかることが面倒な方にもOKです。

最初に緑の音符、次に赤の音符、と動かしていきます。そのようにして入力していくのですが、クイック・ドロウ・スイッチの使用により、一度にたくさんの音符を入力することもできます。音符の長さの調節

は、クリックしたままで、指を押したままで動かせば、指を離れた長さのところまで音符が伸びますので、音が続いて演奏されます。音符の長さがわからなくても「だいたい、これくらいの長さだろう」と指を離すだけで決まりますので、拍数が合わなかったりということはありません。今までのように手を叩き拍数を数え、長さを計算してインプットという、頭の痛くなるような作業から解放されます。

楽譜から入力してみよう

今までの説明は、音符がわからなくても入力できる、という方法です。でも、これでは、音符がわかっている人には、こんな、不便なものはない！といわれてしまいます。

このソフトは画面に鍵盤が表示され、そこで音程を確認できるようにになっています。

ドロウのメニューの「Pitch Guide」で音程の決定の方法を選びます。ここで、「Free」を選べばコード進行にもメロディにも左右されずに演奏されます。「Scale ruler」をonにすると、画面の左はじに鍵盤の絵とアルファベットと数字が表示されます。

これは、スケール、リズム、音程、オクターブの表示です。ここで、赤いドットがマウスの動きと同じように動き、音程を示すので、確実な音程を理解することができます。これまでに、自分で適当に音符を入力していた人は逆に自分の作ったメロディの音程もわかるというわけです。また、演

奏中にコードが画面上部のメニューバーに表示されます。これで、コード進行までもわかってしまいます。楽譜を見ての入力ではクラシックのオーケストラなどが最適です。各パートごとに1音ずつ入力していくのですから、楽譜どおりに音符を分けずに入力します。それを各パートに音色を決めます。ストリングの音など使えば、完璧なオーケストラ気分が味わえます。また、各パートの音量設定を慎重に決めながら、小節ごとに演奏スピードを変更すれば、本格的なオーケストラ演奏になります。でも、それだけの演奏で、まだ物足りないという人は、それにリズムを加えてみてください。「クラシックにドラムが入るの？」と思いついてしまうが、演奏してみると、これがオシャレなアレンジ曲に変身するのです。

広がる楽しさ

どうせなら、作曲した曲をほかの人に聴いてもらえたほうがひとしおに楽しいものです。

そこで、PC-8801FA/MAの本体からライン・アウトしてカセット・デッキに接続して録音すればいいのです。そうすれば、いつでも、どこでも、誰にでも聴いてもらうことができます。

このように、誰にでも簡単に音楽を楽しむことができるのが「Instant Music」なのです。

Let's Enjoy Music!

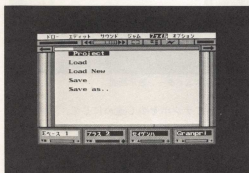


写真5 音符入力のためのメニュー表示

主な添付曲

Bach's Joy

あまりにも有名なバッハのカンタータです。
Mountain King
グreekの「ペールギュント組曲第一」の
アレンジ曲

The Entertainer

映画「スティング」のテーマソング

Home On The Range

アメリカ民謡の「群のわが家」

Kusa Keiba

フオスターの「草競馬」のアレンジ曲

Troika

ロシア民謡の「トロイカ」

Les Champs-Elysees

シャンソンで最も有名で親しまれている、

「おおシャンゼリゼ」

そのほかにもサンバ、ラテンと色々なジャンル、オリジナルが数十曲。

FA/MAの拡張モード

Sec.1 FA/MAのBASIC Sec.2 PCM音源ファイリングシステム「VOICE MEMO 88」

PART 2ではFA/MAに追加されたBASICの拡張モードについて解説する。この機能強化によってSRで拡張されたモードが「拡張モード1」、FA/MAで拡張されたモードが「拡張モード2」ということになる。この拡張モードの増設によりサウンド機能が強化されたわけである。解説のあとにはPCM音源をBASIC命令によりコントロールするファイリングシステム「VOICE MEMO 88」を掲載する。

1. FA/MAのBASIC

FA/MAの登場によって、パーソナルコンピュータとしてのポテンシャルが一段と増大したPC-8800シリーズ。前述したようにFAでは、FM音源チップYM2608に対応してサウンド機能が強化され、MAでは、これに加えて日本語処理機能が強化された。サウンド機能に関してはFA/MAより前の機種でも、オプションのサウンドボードⅡを装着すればFA/MAと同じ機能が実現可能である。

この機能強化に伴って、BASICの拡張命令には、2つのモードが存在することになった。SRで拡張されたモードを拡張モード1、今度の拡張モード2という。

拡張モード2の場合には、FM音源6音、SSG音源3音、リズム音源6音、ADPCM音源1音をフルに発揮でき、プリセットできる音色の数も62個から127個になった。

ただし、音源モードによって同時にプレイできる音には制限があって、今度拡張された音源モード#5では、FM音源6音、リズム音源6音、ADPCM音源1音は同時にプレイできるが、SSG音源3音は使えない。

また、拡張モード2は日本語BASICが使用できず、拡張モード1ではNEW CMDですぐに拡張できたものが、拡張モード2では、プログラムファイル“fm.ipl”を実行しなければならない。

とはいえ、FA/MAになって、オーディオ機能が大幅に強化され、PC-8800シリーズをミュージックマシンとして使うのなら、あとはその機能を十分発揮するようなソフトウェア・アプリケーションの出現次第である。

また、同時にミュージックのほかにも実現できることがたくさんある。パソコン通信、データベース、ワープロ、ゲームなどがそれである。そして、これらの機能をマンマシン・インタフェースによって、本当にパーソナルなものにするキーポイントの1つが、このFA/MAには標準で搭載されている。人間の声をも取り扱うことのできるデジタルサンプリング機能がそれである。

K.I.T.T.のようにコンピュータと対話するのは、まだ、音声認識というような少々手間のかかる機能が必要だが、確実にその第一歩を踏み出していることは間違いない。

BASICの拡張命令

拡張モード1から拡張モード2になって、いったい何が変わったのか。簡単にまとめてみよう。

そもそも拡張命令とは、次の2つからなっている。

①アナログパレット機能

②サウンド機能

①のアナログパレット機能は512色から8色を選んで表示する機能で、拡張モード2になっても変わっていない。したがって、②のサウンド機能が変ったのである。変った部分は追加と新設に分けることができる。

③追加

①モノラルFM音源3音がステレオ6音になった。

②プリセット音色62個が127個使えるようになった。

④新設

①ステレオリズム音源6音が新設された。

②PCM音源1音が新設された。

PCM サンプリング機能の活用性

FA/MAになってオーディオ機能が強化されたわけだが、なかでもPCMサンプリング機能は見逃せない。

サンプリングレートは4から16KHzで256Kバイトの専用メモリ（サウンドメモリ）を持っていて、4KHzで約120秒、16KHzで約30秒のサンプリングが行える。この専用メモリの内容はフロッピーディスクに落とすことができるので、いわば、サンプリングレコーダとしても使用可能である。

最近では、電子レンジやカメラだって喋る時代だが、88は88らしく喋らせてみたい。

そこでこの特長あるサンプリング機能を用いた「声の出るスケジューラ」が次に登場する「VOICE MEMO 88」である。

2. PCM音源 ファイリングシステム 「VOICE MEMO 88」

機能説明

「VOICE MEMO 88」は、「VOICE MEMO」と「TIME MANAGER」から成っている(図1参照)。前者はいわば録音機であり、約1秒から約64秒までの音声データを日時データつきで取り扱うことができる。後者は、その音声、日時データを基にして時間の管理を行うものである。この両者のコンビネーションによって、音声によるスケジュール管理は飛躍的に向上する。

このシステムに対するコマンドは、プログラムの起動を除けば、すべてテンキー側だけで行えるように設計している(図2参照)。フルキーを使うのであれば、このシステムの意味は半減するだろう。音声認識ができるになれば、あえてキーを使う必要もなくなるかもしれない。

●「VOICE MEMO」

VOICE MEMOは、デジタルサンプリング機能を使って音声データを日づけと時間つきでファイルするものであり、録音・再生・消去・プロテクトなどが可能である。

日づけは1987年1月1日から1999年12月31日まで指定でき、閏年も計算の上、日時とともに曜日まで表示してくれる。しかも、音声データを日時について若い順にソートするから録音・消去を何度繰り返しても同じ年、月、日、時、および分が、固まって表示されるという仕組みである。また、ドキュメントとして使用するのなら、日時指定を省略してシステムクロックの内容をそのまま日時データとして用いることもできる。

1つの音声データの録音時間は約1秒から約64秒まで約0.5秒単位で指定可能であり、1枚のフロッピーディスクに異なった秒数の音声データが混在しても構わない。

図1 VOICE MEMO 88
概念図

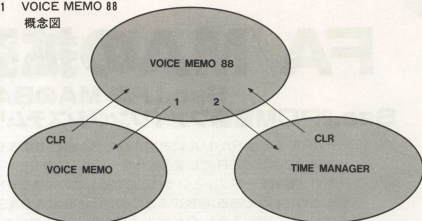
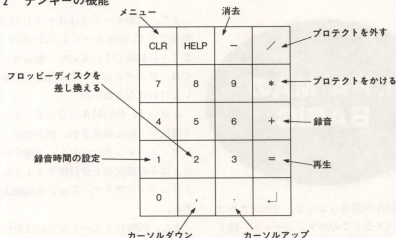


図2 テンキーの機能



ただし、1枚に収録できる音声データの数は16データまでである。ADPCMのサンプリングレートが8 KHzのとき、2Dのフロッピーディスクは、およそ4ないし5秒×16データでいっぱいになる。そこで、「VOICE MEMO 88」起動時の録音時間の初期値は約4.5秒(TIME=45)に設定しておいた。ちょっととしたメモなら、このくらいがちょうどいい。

音声データファイルの管理はディレクトリファイル「VOICE.DIR」が行っている。これが最初に作られるのは、セットした音声データ用のフロッピーディスクに「VOICE.DIR」が存在しないときである。音声データ用フロッピーディスクは、なるべく専用のものを用意するのがよい。

VOICE MEMO 88システムが作成するファイルは、この「VOICE.DIR」と音声データファイル「V10000.PCM」から「V10015.PCM」までの計17ファイルだけである。これらのファイルをVOICE MEMO 88シ

ステム以外から操作してはならない。

●「TIME MANAGER」

TIME MANAGERはシステムクロックの日時がVOICE MEMOでファイルした日時と一致したときに、自動的にその音声データを発生させ、その時刻がきたことを音声の情報で知らせてくれる。

- 一致条件は、
①日時のモード (DATE AND TIME)
②時間だけのモード (TIME ONLY)
を選択することができる。

①の日時のモードを選択すれば、その日づけかつその時刻がきたときに知らせてくれるし、②の時間だけのモードを選択すれば日づけに関係なく、その時刻ごとに知らせてくれる。音声データ用フロッピーディスクにファイルした日時や、TIME MANAGERの用途に応じて選ばばよい。

イヤリー、マンスリー、およびウィークリーな情報であれば①の日時のモードを、また、デイリーなものであれば②の時間だ

けのモードが適当であろう。

●テレビ番組をマネージメント

たとえば、自分の見たいテレビ番組をマネージメントする方法を考えてみよう。テレビ番組のタイトルとチャンネルなどを番組の開始時刻でファイルして、TIME MANAGER を②の時間だけのモードで実行すれば、あとはほかの仕事に没頭していても番組開始時刻がくれば何の番組が何チャンネルで放送されるのかが、ちゃんと音声で耳の中に飛び込んでくる。だいたいテレビ番組はウィークリーなものが多いので、曜日ごとに音声データ用フロッピーディスクを用意すればよい。1週間に見たい番組の数がフロッピーディスク1枚に納まるのであれば、週の初めにデータを設定しておいて①の日時のモードで起動することもできる。

動作環境

VOICE MEMO 88を運用するには次のシステムが必要である。

- ① PC8801FA/MA, または,
SR以降+サウンドボードII

この中にはN88-BASIC Ver2.0以上と、それを拡張モード2にするプログラムファイルが含まれている

- ② ディスプレイ

- ③ アンプ+スピーカー+マイク

コンピュータ本体と、これらを適当なコードで接続する。ラジカセかミニコンボがあると便利である

システムディスクの作成

- ① BASIC (拡張モード2) のシステムディスクを作成する

サウンドボードIIなら、付属のシステムディスクをそのままコピーすればよいし、FA/MAなら、システムディスクをコピーした後、リスト1がセーブできるように不要なファイルをいくつか削除すればよい

- ② IDセクタを次のように書き換える

- 1 同時にオープンするファイル数=1
2 西暦の下2ケタ=88
3 BASICプログラム
BLOAD "fm. ip1", R

- ③ リスト1のBASICプログラムを入力して次のようにフロッピーディスクにセーブする

SAVE "VM88" ◎

(注) 1ドライブシステムの場合はセーブの前にBASICプログラム中のDRIVES="2"をDRIVES="1"に書き換えておく。

起動方法

- ① コンピュータの電源スイッチを入れ、

DRIVE 1にVOICE MEMO 88のシステムディスクをセットしてリセットスイッチを押す

- ② "カクチョウネイロヲシヨウシマスカ
[Y/N]"に対して[N]キーを押す

- ③ RUN "VM88" ◎でVOICE MEMO 88が起動する

操作方法

VOICE MEMO 88を起動すると、まずシステムクロックの設定を行う。現在の日づけと時間を入力するが、そのままでよいときは、単にRETURNキーを押す。もう一度、日づけと時間を表示して確認してするので、再設定ならばHELPキーを押す、よければRETURNキーでメインメニューに移る。

メインメニューでは、[1]キーでVOICE MEMO、[2]キーでTIME MANAGER、[3]キーでシステムクロックの設定に制御が移る。[9]キーを押すと終了である。

●「VOICE MEMO」

最初に、

"SET PCM DATA DISK

ON DRIVE 2

OK (RETURN) ?"

と表示して、音声データ用フロッピーディスクのセットを要求してくるので、セットしたらRETURNキーを押す。以後、音声データ用フロッピーディスクの差し替えは、必ずこの表示のときだけに行わなければならない。

機能ごとのキーの使用方法是、

- ① [1] または [9]

データ選択用のカーソルを上げる

- ② [2] または [3]

データ選択用のカーソルを下げる

- ③ [4]

録音を行う
まず、

RETURN] REC START

と表示され、RETURNキーを押した直後から録音が開始される。この時点では、録音データはサウンドメモリ内に蓄えられる。

終了すれば、

- 1] OK 2] TRY AGAIN

と表示される。

- ④ [1] キーを押すと、

SET DATE OR TIME ?

- 1] YES 2] PRESENT

と表示され、[1]キー押すことで日時を強制的に設定することができるが、システムクロックの値でよければ[2]キーを押せばよい。

以上の操作が終わると、サウンドメモリ内のデータをディスクに落とす。

なお、録音時間は、あらかじめ[1]キーで設定しておく。

- ④ [4] キー 再生

音声データ用フロッピーディスクからサウンドメモリに指定したデータを読み込み、そのデータを録音した時間だけ再生を行う。

- ⑤ [5] キー 消去

指定したデータの消去を行う。ただし、プロテクトのかかったデータは消去できない。

- ⑥ [6] キー プロテクトをかける

指定したデータにプロテクトをかける。以後、[7]キーでプロテクトを外すまで、[6]キーでは消去できない。

ただし、そのフロッピーディスクのフォーマットを行ったり、VOICE MEMO 88のシステム以外で操作すれば、このかぎりではない。

- ⑦ [7] キー プロテクトを外す

指定したデータのプロテクトを外す。

- ⑧ [1] キー 録音時間の設定

録音したい時間を10から640の範囲で設定する。1は約0.1秒に相当する。

- ⑨ [2] キー フロッピーディスクを差し替える

音声データ用フロッピーディスクの差し替えを行う。

- ⑩ [CLR] キー メインメニューに移る

VOICE MEMO 88のメインメニュー

に移る。

次に、一致条件のモードセレクトを行う

るデータの日時を表示する。

●「TIME MANAGER」

と、マネージメントを開始する。

VOICE MEMO と同様に、最初に音声デ

画面には、システムクロックの日時をリ

ータ用フロッピーディスクをセットする。

アルタイムに表示し、その下に次に発生す

リスト1

```
1000 '*****
1010 ' VOICE MEMO 88 Version 1.0
1020 ' Copyright 1987 H.BANDO All rights reserved.
1030 ' 1987 12.17 file name "VM88"
1040 '*****
1050 '
1060 ON ERROR GOTO *ERROR.TRAP
1070 ON HELP GOSUB *HP:HELP ON
1080 ON STOP GOSUB *SP:STOP ON
1090 ON KEY GOSUB *KY:KEY ON
1100 '
1110 DEFINT A-Z
1120 CLS 3:SCREEN 0,0:COLOR 7
1130 WIDTH 40,25:CONSOLE 0,25,0,1
1140 '
1150 DEF USR0=&H4290 ' cursor on
1160 DEF USR1=&H428B ' cursor off
1170 SF=8 ' sampling frequency
1180 TS=0:TR=45:TP=45:MAX.C=750
1190 CMD PCM 255,3,SF,0
1200 CMD.BGM 1:CMD PLAY #5,""
1210 FOR I=0 TO 7:COLOR=(I,1):NEXT I
1220 LINE(0,0)-(639,199),1,BF
1230 '
1240 N=15
1250 DIM F$(N),F1$(N),FS$(N)
1260 DIM N$(N),N1$(N),NS$(N)
1270 DIM V$(N),V1$(N),VS$(N)
1280 DIM K$(N),K1$(N),KS$(N)
1290 DF=0 ' dir flag found=1
1300 DRIVES="2":DNAME$="VOICE.DIR":DIR$=DRIVE$+"":+DNAME$
1310 '
1320 IY=1987:IOFF=3 '(0~6) 1987/01/01=THU
1330 DIM Y(1,12)
1340 Y(0,0)=0:Y(0,1)=31:Y(0,2)=28:Y(0,3)=31:Y(0,4)=30
1350 Y(0,5)=31:Y(0,6)=30:Y(0,7)=31:Y(0,8)=31:Y(0,9)=30
1360 Y(0,10)=31:Y(0,11)=30:Y(0,12)=31
1370 Y(1,0)=0:Y(1,1)=31:Y(1,2)=29:Y(1,3)=31:Y(1,4)=30
1380 Y(1,5)=31:Y(1,6)=30:Y(1,7)=31:Y(1,8)=31:Y(1,9)=30
1390 Y(1,10)=31:Y(1,11)=30:Y(1,12)=31
1400 DIM WK$(6)
1410 WK$(0)="SUN":WK$(1)="MON":WK$(2)="TUE":WK$(3)="WED"
1420 WK$(4)="THU":WK$(5)="FRI":WK$(6)="SAT"
1430 '
1440 *INIT.DATE.TIME
1450 CLS
1460 LOCATE 3,4:PRINT "-----| SET SYSTEM CLOCK |-----"
1470 LOCATE 4,6:PRINT "YEAR= 1987 ~ 1999 [ 87 ~ 99 ]"
1480 LOCATE 9,8:PRINT "HELP] TRY AGAIN"
1490 LOCATE 9,10:PRINT "RETURN] DEFAULT"
1500 ON HELP GOSUB *HP.IT:HELP ON
1510 *INIT0
1520 LOCATE 7,14:PRINT "PRESENT DATE = ";DATE$
1530 LOCATE 7,16:LINE INPUT "DATE ##/##/## ? ";DT$
1540 IF DT$="" THEN *INIT1
1550 IF LEN(DT$)<>8 THEN *INIT.DATE.TIME
1560 GOSUB *GET.NAME
1570 GOSUB *GET.YMD
1580 GOSUB *CHECK.DATE:IF F=0 THEN *INIT.DATE.TIME
1590 DATE$=DT$
1600 '
1610 *INIT1
1620 LOCATE 7,18:PRINT "PRESENT TIME = ";TIME$
1630 LOCATE 7,20:LINE INPUT "TIME ##.##.## ? ";TM$
1640 IF TM$="" THEN *INIT2
1650 IF LEN(TM$)<>8 THEN *INIT.DATE.TIME
1660 GOSUB *GET.NAME
1670 GOSUB *GET.TIME
1680 GOSUB *CHECK.TIME:IF F=0 THEN *INIT.DATE.TIME
1690 TM$=MID$(TM$,1,2)+":"+MID$(TM$,4,2)+":"+MID$(TM$,7,2)
```



```

1700 TIME$=TM$
1710 HELP OFF
1720 '
1730 *INIT2
1740 CLS
1750 WHILE 1
1760   LOCATE 7,10:PRINT "PRESENT DATE  = ";DATE$
1770   LOCATE 7,12:PRINT "PRESENT TIME  = ";TIME$
1780   LOCATE 7,16:PRINT "OK ( RETURN / HELP ) ? ";
1790   A$=INKEY$
1800   IF A$=CHR$(13) THEN *MAIN
1810 WEND
1820 '
1830 *MAIN
1840 GOSUB *FAN
1850 ON HELP GOSUB *HP :HELP ON
1860 GOSUB *DSP.MENU1
1870 GOSUB *KEYIN :PRINT A$
1880 IF A$="1" THEN *VEDITO
1890 IF A$="2" THEN *TM.MANAGE
1900 IF A$="3" THEN *INIT.DATE.TIME
1910 IF A$="9" THEN *QUIT
1920 GOTO *MAIN
1930 '
1940 *VEDITO
1950 GOSUB *DSP.MENU0
1960 GOSUB *KEYIN
1970 IF A$<>CHR$(8) THEN *VEDITO
1980 GOSUB *DSP.CHECK.DIR
1990 GOSUB *SET.DIR
2000 '
2010 *VEDIT
2020 GOSUB *GET.DIR
2030 FOR I=0 TO N
2040   FS$(I)=F$(I):NS$(I)=N$(I):VS$(I)=V$(I):KS$(I)=K$(I)
2050 NEXT I
2060 GOSUB *DIR.SORT
2070 GOSUB *DSP.EDIT
2080 NF=YC:GOSUB *GET.POS:PRINT A$
2090 GOSUB *CNV.V ' out yy v$
2100 IF A$="+" THEN GOSUB *REC:GOTO *VEDIT
2110 IF A$="=" THEN GOSUB *PLY:GOTO *VEDIT
2120 IF A$="-" THEN GOSUB *ERS:GOTO *VEDIT
2130 IF A$="/" THEN GOSUB *UNP:GOTO *VEDIT
2140 IF A$="*" THEN GOSUB *PRT:GOTO *VEDIT
2150 IF A$="1" THEN GOSUB *CHANGE.T:GOTO *VEDIT
2160 IF A$="2" THEN *VEDITO
2170 IF A$=CHR$(12) THEN *MAIN
2180 GOTO *VEDIT
2190 '
2200 *FAN
2210 CMD PLAY "@44V1303L32CC+L1D"
2220 RETURN
2230 '
2240 *FAN1
2250 CMD PLAY "@27V1304L16CEGB>L1C"
2260 RETURN
2270 '
2280 *DSP.MENU0
2290 CLS
2300 LOCATE 6,10:PRINT "SET PCM DATA DISK ON DRIVE ";DRIVE$
2310 LOCATE 10,12:PRINT "OK ( RETURN ) ? ";
2320 RETURN
2330 '
2340 *DSP.MENU1
2350 CLS
2360 LOCATE 2,4:PRINT "——| VOICE MEMO 88 VERSION 1.0 |——"
2370 LOCATE 12,8 :PRINT "1) VOICE MEMO"
2380 LOCATE 12,10:PRINT "2) TIME MANAGER"
2390 LOCATE 12,12:PRINT "3) SET SYSTEM CLOCK"
2400 LOCATE 12,16:PRINT "9) END"
2410 LOCATE 12,20:PRINT "COMMAND ";
2420 RETURN
2430 '
2440 *DSP.EDIT
2450 CLS
2460 LOCATE 1,1:PRINT "——| VOICE MEMO |——"
2470 XD=4 :YD=3 :YC=0
2480 FOR II=0 TO N
2490   IF FS$(II)="01" OR FS$(II)="81" THEN *DVD1 ELSE *DVD2

```

```

2500 *DVD1
2510 Y$=RIGHT$(STR$(YC),LEN(STR$(YC))-1)
2520 Y$=RIGHT$("00"+Y$,2)
2530 YM$=RIGHT$(STR$(YC+1),LEN(STR$(YC+1))-1)
2540 YM$=RIGHT$("00"+YM$,2)
2550 N$=NS$(1):GOSUB *DIV.NAME
2560 N$=NS$(1):GOSUB *GET.YMD
2570 GOSUB *GET.WEEK:IF F=0 THEN *ER2
2580 IF FS$(1)="81" THEN CSR$="*" ELSE CSR$="O"
2590 TG$=RIGHT$(K$(1),3)
2600 LOCATE XD,YD:PRINT YM$,";";" ";DT$;" ";WEEK$;" ";CSR$;" ";TM$;" ";TG$
2610 YD=YD+1:YC=YC+1
2620 *DVD2
2630 NEXT I1
2640 LOCATE 1,20:PRINT "[+]REC [=]PLAY [-]ERASE [*]PRO [/]UNPR"
2650 LOCATE 1,21:PRINT "[1]TIME( [2]CH.DISK [CLR]MENU"
2660 LOCATE 9,21:PRINT TR;")"
2670 RETURN
2680 '
2690 *DSP.CHECK.DIR
2700 CLS
2710 LOCATE 7,11:PRINT "CHECK VOICE DIRECTORY ...."
2720 RETURN
2730 '
2740 *DSP.MAKE.DIR
2750 CLS
2760 GOSUB *FAN1
2770 LOCATE 7,11:PRINT "NOW MAKING VOICE.DIR ...."
2780 RETURN
2790 '
2800 *KEYIN
2810 A$=INKEY$:A1$=USRO(A1$)
2820 IF A$="" THEN *KEYIN
2830 RETURN
2840 '
2850 *QUIT
2860 STOP OFF:HELP OFF:KEY OFF:TIME$ OFF
2870 FOR I=0 TO 7:COLOR=(I,1):NEXT I
2880 WIDTH 80,25:CONSOLE ,,1:CLS 3
2890 ON ERROR GOTO 0
2900 END
2910 '
2920 *SP:GOTO *QUIT
2930 *HP
2940 *KY
2950 RETURN
2960 '
2970 *SET.DIR
2980 DM$=ATTR$(DIR$) ' check dir
2990 RETURN
3000 '
3010 *ERROR.TRAP
3020 IF ERR=53 AND DF=0 THEN *MAKE.DIR ELSE *ER1
3030 '
3040 *MAKE.DIR
3050 GOSUB *DSP.MAKE.DIR
3060 FOR I=0 TO N
3070 DD$=RIGHT$(STR$(I),LEN(STR$(I))-1)
3080 F$(I)="00":N$(I)="000000000000":V$(I)=RIGHT$("00"+DD$,2)
3090 K$(I)="0000"
3100 NEXT I
3110 '
3120 GOSUB *PUT.DIR
3130 DF=1
3140 RESUME NEXT
3150 '
3160 *ER1
3170 CLS 3:STOP OFF:HELP OFF:KEY OFF:TIME$ OFF
3180 FOR I=0 TO 7:COLOR=(I,1):NEXT I
3190 WIDTH 80,25:CONSOLE ,,1
3200 ON ERROR GOTO 0
3210 END
3220 '
3230 *ER2
3240 CLS 3:STOP OFF:HELP OFF:KEY OFF:TIME$ OFF
3250 FOR I=0 TO 7:COLOR=(I,1):NEXT I
3260 WIDTH 80,25:CONSOLE ,,1
3270 PRINT "SYSTEM ERROR 100"
3280 ON ERROR GOTO 0
3290 END

```

```

3300 '
3310 *PUT.DIR
3320 OPEN DIR$ AS #1
3330 FOR I=0 TO N
3340 FIELD #1,I*16 AS DMY$,2 AS F1$(I),12 AS N1$(I),2 AS V1$(I)
3350 NEXT I
3360 FOR I=0 TO N
3370 RSET F1$(I)=F$(I):RSET N1$(I)=N$(I):RSET V1$(I)=V$(I)
3380 NEXT I
3390 PUT #1,1
3400 FOR I=0 TO N
3410 FIELD #1,I*4 AS DMY$,4 AS K1$(I)
3420 NEXT I
3430 FOR I=0 TO N
3440 RSET K1$(I)=K$(I)
3450 NEXT I
3460 PUT #1,2
3470 CLOSE #1
3480 RETURN
3490 '
3500 *GET.DIR
3510 OPEN DIR$ AS #1
3520 FOR I=0 TO N
3530 FIELD #1,I*16 AS DMY$,2 AS F1$(I),12 AS N1$(I),2 AS V1$(I)
3540 NEXT I
3550 GET #1,1
3560 FOR I=0 TO N
3570 F$(I)=F1$(I):N$(I)=N1$(I):V$(I)=V1$(I)
3580 NEXT I
3590 FOR I=0 TO N
3600 FIELD #1,I*4 AS DMY$,4 AS K1$(I)
3610 NEXT I
3620 GET #1,2
3630 FOR I=0 TO N
3640 K$(I)=K1$(I)
3650 NEXT I
3660 CLOSE #1
3670 RETURN
3680 '
3690 *DIR.SORT
3700 ' in N FS$(N) NS$(N) VS$(N) KS$(N)
3710 ' use W0$ W1$ W2$ W3$ NW I I *B1
3720 FOR I=N TO 1 STEP -1
3730 NW=I-1
3740 W0$=FS$(I-1):W1$=NS$(I-1):W2$=VS$(I-1):W3$=KS$(I-1)
3750 FOR J=I TO N
3760 IF W1$ <= NS$(J) THEN *B1
3770 FS$(J-1)=FS$(J):NS$(J-1)=NS$(J):VS$(J-1)=VS$(J):KS$(J-1)=KS$(J)
3780 NEXT J
3790 *B1
3800 IF NW <> J-1 THEN FS$(J-1)=W0$:NS$(J-1)=W1$:VS$(J-1)=W2$:KS$(J-1)=W3$
3810 NEXT I
3820 RETURN
3830 '
3840 *GET.NAME
3850 ' in DT$ TMS
3860 ' out Ns
3870 DT1$=MID$(DT$,1,2)+MID$(DT$,4,2)+MID$(DT$,7,2)
3880 TMS1$=MID$(TMS,1,2)+MID$(TMS,4,2)+MID$(TMS,7,2)
3890 Ns=DT1$+TMS1$
3900 RETURN
3910 '
3920 *DIV.NAME
3930 ' in N$
3940 ' out DT$ TMS$
3950 DT$=MID$(N$,1,2)+"/"+MID$(N$,3,2)+"/"+MID$(N$,5,2)
3960 TMS$=MID$(N$,7,2)+":"+MID$(N$,9,2)+":"+MID$(N$,11,2)
3970 RETURN
3980 '
3990 *MAKE.FILE
4000 ' in V$
4010 ' out SNAME$
4020 SNAME$=DRIVE$+" ":"V10"+RIGHT$("000"+V$,3)+" .PCM"
4030 RETURN
4040 '
4050 *GET.POS
4060 ' in YD NF (0~N+1)
4070 ' out YY
4080 IF NF>N THEN Y=YD-1 ELSE Y=YD
4090 X=21:YI=3

```



```

4100 *GP1
4110 LOCATE X,Y
4120 GOSUB *KEYIN
4130 IF NF=0 THEN *GP2
4140 IF A$=CHR$( &H1E) THEN Y=Y-1
4150 IF A$=CHR$( &H1F) THEN Y=Y+1
4160 IF A$=CHR$( &H2C) THEN Y=Y-1
4170 IF A$=CHR$( &H2E) THEN Y=Y+1
4180 IF Y < Y1 THEN Y=NF+Y1
4190 IF Y > NF+Y1 THEN Y=Y1
4200 *GP2
4210 IF A$="+" THEN *GP3
4220 IF A$="-" THEN *GP3
4230 IF A$="*" THEN *GP3
4240 IF A$="/" THEN *GP3
4250 IF A$="1" THEN *GP3
4260 IF A$="2" THEN *GP3
4270 IF A$=CHR$(12) THEN *GP3
4280 GOTO *GP1
4290
4300 '
4310 *GP3
4320 YY=CSRLIN-YY+1
4330 RETURN
4340 '
4350 *CNV.V
4360 ' in N YY (0~N+1)
4370 ' out V$ V
4380 IF YY=0 THEN *CV0
4390 IF YY-1 > N THEN *VOICE.FULL
4400 YI=0 ' yy non zero
4410 FOR I=0 TO N
4420 IF FS$(I)="01" OR FS$(I)="81" THEN *CV1 ELSE *CV2
4430 *CV1
4440 YI=YI+1
4450 IF YY=YI THEN *CV3
4460 *CV2
4470 NEXT I
4480 GOTO *CV0
4490 *CV3
4500 V$=VS$(I):V=VAL(V$)
4510 RETURN
4520 '
4530 *CV0 ' yy zero
4540 FOR I=0 TO N
4550 IF FS$(I)<>"01" AND FS$(I)<>"81" THEN *CV4
4560 NEXT I
4570 *CV4
4580 V$=VS$(I):V=VAL(V$)
4590 RETURN
4600
4610 *UNP
4620 IF F$(V)<>"01" AND F$(V)<>"81" THEN *UNP1
4630 GOSUB *CLS.M
4640 LOCATE 3,23:PRINT "UN-PROTECT ...";
4650 F$(V)="01"
4660 GOSUB *PUT.DIR
4670 GOSUB *CLS.M
4680 RETURN
4690 '
4700 *UNP1
4710 GOSUB *CLS.M
4720 LOCATE 3,23:PRINT "CAN NOT UN-PROTECT !";
4730 FOR I=0 TO 2000:NEXT I
4740 GOSUB *CLS.M
4750 RETURN
4760 '
4770 *PRT
4780 IF F$(V)<>"01" AND F$(V)<>"81" THEN *PRT1
4790 GOSUB *CLS.M
4800 LOCATE 3,23:PRINT "PROTECT ...";
4810 F$(V)="81"
4820 GOSUB *PUT.DIR
4830 GOSUB *CLS.M
4840 RETURN
4850 '
4860 *PRT1
4870 GOSUB *CLS.M
4880 LOCATE 3,23:PRINT "CAN NOT PROTECT !";
4890 FOR I=0 TO 2000:NEXT I

```

```

4900 GOSUB *CLS. M
4910 RETURN
4920 '
4930 *ERS
4940 IF F$(V)<>"01" AND F$(V)<>"81" THEN *ES1
4950 IF F$(V)="81" THEN *ES2
4960 GOSUB *CLS. M
4970 LOCATE 3,23:PRINT "ERASE ...";
4980 F$(V)="02":K$(V)="00"
4990 GOSUB *PUT. DIR
5000 GOSUB *MAKE. FILE
5010 KILL SNAME$
5020 GOSUB *CLS. M
5030 RETURN
5040 '
5050 *ES1
5060 GOSUB *CLS. M
5070 LOCATE 3,23:PRINT "CAN NOT ERASE !";
5080 FOR I=0 TO 2000:NEXT I
5090 GOSUB *CLS. M
5100 RETURN
5110 '
5120 *ES2
5130 GOSUB *CLS. M
5140 LOCATE 3,23:PRINT "PROTECT VOICE !";
5150 FOR I=0 TO 2000:NEXT I
5160 GOSUB *CLS. M
5170 RETURN
5180 '
5190 *PLY
5200 IF F$(V)<>"01" AND F$(V)<>"81" THEN *PL1
5210 GOSUB *CLS. M
5220 LOCATE 3,23:PRINT "LOADING ...";
5230 GOSUB *MAKE. FILE
5240 CMD PCM LOAD SNAME$,TS
5250 TP=VAL(K$(V))
5260 GOSUB *CLS. M
5270 LOCATE 3,23:PRINT "PLAYING":TP;" ...";
5280 CMD PCM PLAY TS,TP
5290 GOSUB *CLS. M
5300 RETURN
5310 '
5320 *PL1
5330 GOSUB *CLS. M
5340 LOCATE 3,23:PRINT "NOT FOUND !";
5350 FOR I=0 TO 2000:NEXT I
5360 GOSUB *CLS. M
5370 RETURN
5380 '
5390 *REC
5400 DFR=DSKF(VAL(DRIVE$))*5
5410 IF DFR < TR THEN *VOICE. FULL
5420 TT=0
5430 FOR I=0 TO N
5440 TT=TT+VAL(K$(I))
5450 NEXT I
5460 IF TT+TR>MAX.C THEN *VOICE. FULL
5470 IF F$(V)="81" THEN *REC5
5480 *REC0
5490 GOSUB *CLS. M
5500 LOCATE 1,23:PRINT "RETRUN REC START TIME =":TR;" ";
5510 GOSUB *KEYIN
5520 IF A$<>CHR$(13) THEN *REC0
5530 GOSUB *CLS. M
5540 LOCATE 3,23:PRINT "RECORDING ..... ";
5550 COLOR=(1,2)
5560 CMD PCM RECORD TS,TR
5570 COLOR=(1,1)
5580 *REC1
5590 GOSUB *CLS. M
5600 LOCATE 3,23:PRINT "1) OK 2) CANCEL ";
5610 GOSUB *KEYIN:PRINT A$;
5620 IF A$="1" THEN *REC2
5630 IF A$="2" THEN *VEDIT
5640 GOTO *REC1
5650 *REC2
5660 GOSUB *CLS. M
5670 LOCATE 1,23:PRINT "SET DATE OR TIME ? 1)SET 2)PRESENT ";
5680 GOSUB *KEYIN:PRINT A$;
5690 IF A$="1" THEN GOSUB *SET.DT :GOTO *REC4

```

```

5700 IF A$="2" THEN *REC3
5710 GOTO *REC2
5720 *REC3
5730 GOSUB *CLS.M
5740 LOCATE 3,23:PRINT "SAVING ..... ";
5750 DT$=DATE$:TM$=TIME$:GOSUB *GET.NAME
5760 *REC4
5770 S$=RIGHT$(STR$(TR),LEN(STR$(TR))-1)
5780 F$(V)="01":N$(V)=N$:K$(V)=RIGHT$("0000"+S$,4)
5790 GOSUB *PUT.DIR
5800 GOSUB *MAKE.FILE
5810 CMD PCM SAVE SNAME$,TS,TR
5820 GOSUB *CLS.M
5830 RETURN
5840 '
5850 *REC5
5860 GOSUB *CLS.M
5870 LOCATE 3,23:PRINT "PROTECT VOICE !";
5880 FOR I=0 TO 2000:NEXT I
5890 GOSUB *CLS.M
5900 RETURN
5910 '
5920 *SET.DT ' help try again
5930 CLS
5940 LOCATE 4,2:PRINT "-----| SET DATE OR TIME |-----"
5950 LOCATE 4,6:PRINT "YEAR= 1987 ~ 1999 [ 87 ~ 99 ]"
5960 LOCATE 4,8:PRINT "      HELP] TRY AGAIN"
5970 LOCATE 4,10:PRINT "      RETURN] DEFAULT"
5980 ON HELP GOSUB *HP.DT:HELP ON
5990 LOCATE 4,14:LINE INPUT "INPUT DATE ##/##/## ";DT$
6000 IF DT$="" THEN DT$=DATE$
6010 IF LEN(DT$)<>8 THEN *SET.DT
6020 GOSUB *GET.NAME
6030 GOSUB *GET.YMD
6040 GOSUB *CHECK.DATE:IF F=0 THEN *SET.DT
6050 '
6060 LOCATE 4,16:LINE INPUT "INPUT TIME ##.##.## ";TM$
6070 IF TM$="" THEN TM$=TIME$
6080 IF LEN(TM$)<>8 THEN *SET.DT
6090 GOSUB *GET.NAME
6100 GOSUB *GET.TIME
6110 GOSUB *CHECK.TIME:IF F=0 THEN *SET.DT
6120 '
6130 HELP OFF
6140 LOCATE 12,20:PRINT "SAVING ..... ";
6150 GOSUB *GET.NAME
6160 RETURN
6170 '
6180 *CHECK.DATE
6190 ' in Y M D
6200 ' out F=0 then error
6210 IF Y<1Y THEN F=0:RETURN
6220 IF M<1 OR M>12 THEN F=0:RETURN
6230 Z=Y:GOSUB *CHECK.LEAP
6240 IF D<1 OR D>Y(LEAP,M) THEN F=0:RETURN
6250 F=1
6260 RETURN
6270 '
6280 *CHECK.TIME
6290 ' in HOR MIN SEC
6300 ' out F=0 then error
6310 IF HOR<0 OR HOR>23 THEN F=0:RETURN
6320 IF MIN<0 OR MIN>59 THEN F=0:RETURN
6330 IF SEC<0 OR SEC>59 THEN F=0:RETURN
6340 F=1
6350 RETURN
6360 '
6370 *HP.DT
6380 RETURN *SET.DT
6390 '
6400 *HP.CT
6410 RETURN *CHANGE.T
6420 '
6430 *HP.IT
6440 RETURN *INIT.DATE.TIME
6450 '
6460 *CLS.M
6470 LOCATE 0,23:PRINT STRING$(38," ");
6480 RETURN
6490 '

```



```

6500 *DSP. VOICE. FULL
6510 CLS
6520 LOCATE 8,6:PRINT "**** VOICE FULL ! ****"
6530 LOCATE 10,10:PRINT "1) VOICE MEMO"
6540 LOCATE 10,12:PRINT "2) CHANGE DATA DISK"
6550 LOCATE 12,16:PRINT "COMMAND ";
6560 RETURN
6570 '
6580 *VOICE. FULL
6590 GOSUB *DSP. VOICE. FULL
6600 GOSUB *KEYIN:PRINT A$
6610 IF A$="1" THEN *VEDIT
6620 IF A$="2" THEN *VEDIT0
6630 GOTO *VOICE. FULL
6640 '
6650 *GET. YMD
6660 ' in N$
6670 ' out Y M D
6680 E$=MID$(N$, 1, 2):Y=VAL(E$)
6690 Y=Y+1900
6700 M$=MID$(N$, 3, 2):M=VAL(M$)
6710 D$=MID$(N$, 5, 2):D=VAL(D$)
6720 RETURN
6730 '
6740 *GET. TIME
6750 ' in N$
6760 ' out HOR MIN SEC
6770 HOR$=MID$(N$, 7, 2):HOR=VAL(HOR$)
6780 MIN$=MID$(N$, 9, 2):MIN=VAL(MIN$)
6790 SEC$=MID$(N$, 11, 2):SEC=VAL(SEC$)
6800 RETURN
6810 '
6820 *GET. WEEK
6830 ' in Y,M,D
6840 ' out WEEK$ F
6850 '
6860 GOSUB *CHECK. DATE :IF F=0 THEN RETURN
6870 '
6880 DAY!=D+IOFF
6890 Z=Y:GOSUB *CHECK. LEAP
6900 FOR I=1 TO M-1
6910 DAY!=DAY!+Y(LEAP,I)
6920 NEXT I
6930 FOR I=1 TO Y-IY
6940 Z=I+IY-1:GOSUB *CHECK. LEAP
6950 DAY!=DAY!+365+LEAP
6960 NEXT I
6970 DAY!=INT(DAY!/7)
6980 DAYO!=DAY!-DAY!*7
6990 WEEK$=WK$(DAYO!)
7000 F=1
7010 RETURN
7020 '
7030 *CHECK. LEAP
7040 ' in Z=YEAR
7050 IF (Z MOD 4)=0 AND (Z MOD 100)<>0 OR (Z MOD 400)=0 THEN LEAP=1:RETURN
7060 LEAP=0
7070 RETURN
7080 '
7090 *CHANGE. T
7100 ON HELP GOSUB *HP.CT:HELP ON
7110 CLS
7120 FS=DSKF(VAL(DRIVE$))*5
7130 TT=0
7140 FOR I=0 TO N
7150 TT=TT+VAL(K$(I))
7160 NEXT I
7170 FT=MAX.C-TT:IF FT<0 THEN FT=0
7180 IF FT<FS THEN FS=FT
7190 LOCATE 5,4:PRINT "— INPUT REC TIME 10~640 —"
7200 LOCATE 10,6:PRINT "DIRVE ";DRIVE$;" :FS:"FREE";
7210 LOCATE 11,10:PRINT "HELPJ TRY AGAIN"
7220 LOCATE 11,12:PRINT "RETURNJ TIME=45"
7230 LOCATE 3,17:INPUT "REC TIME ( 10 ~ 640 STEP 5 ) ";T$
7240 T=VAL(T$)
7250 IF T$="" THEN T=45
7260 IF T<10 OR T>640 THEN *CHANGE. T
7270 TR=T-(T MOD 5)
7280 HELP OFF
7290 RETURN

```

```

7300 '
7310 *TM. MANAGE
7320 TIME$ OFF
7330 GOSUB *DSP. MENUO
7340 GOSUB *KEYIN
7350 IF A$<>CHR$(13) THEN *TM. MANAGE
7360 '
7370 WHILE 1
7380 CLS
7390 LOCATE 6,6:PRINT "-----| MODE SELECT |-----"
7400 LOCATE 10,10:PRINT "1) DATE AND TIME"
7410 LOCATE 10,12:PRINT "2) TIME ONLY"
7420 LOCATE 14,16:PRINT "COMMAND ";
7430 GOSUB *KEYIN:PRINT A$
7440 IF A$="1" THEN OTF=0:GOTO *EXT. SEL
7450 IF A$="2" THEN OTF=1:GOTO *EXT. SEL
7460 WEND
7470 *EXT. SEL
7480 '
7490 GOSUB *GET. DIR
7500 FOR I=0 TO N
7510 FS$(I)=F$(I):NS$(I)=N$(I):VS$(I)=V$(I):KS$(I)=K$(I)
7520 NEXT I
7530 '
7540 IF OTF=0 THEN *OTFO
7550 FOR I=0 TO N
7560 NS$(I)="?????" + RIGHT$(NS$(I), 6)
7570 NEXT I
7580 *OTFO
7590 '
7600 GOSUB *DIR. SORT
7610 GOSUB *TM. SETON
7620 TIME$ ON
7630 '
7640 CLS
7650 LOCATE 6,6:PRINT "-----| TIME MANAGER |-----"
7660 LOCATE 13,16:PRINT "CLR) TO MENU"
7670 WHILE 1
7680 LOCATE 6,10:PRINT "PRESENT=";DATE$;" ";TIME$
7690 LOCATE 6,12:PRINT "NEXT =";DT$;" ";TM$
7700 A$=INKEY$
7710 IF A$=CHR$(12) THEN TIME$ OFF:GOTO *MAIN
7720 WEND
7730 '
7740 *TIME. T
7750 TIME$ OFF
7760 YY=TN+1:GOSUB *CNV. V
7770 GOSUB *MAKE. FILE
7780 CMD PCM LOAD SNAME$,TS
7790 TP=VAL(K$(V))
7800 CMD PCM PLAY TS,TP
7810 GOSUB *TM. SETON
7820 TIME$ ON
7830 RETURN
7840 '
7850 *TM. SETON
7860 TM$=TIME$
7870 DT$=DATE$
7880 GOSUB *GET. NAME
7890 IF OTF=1 THEN NT$="?????" + RIGHT$(N$, 6) ELSE NT$=N$
7900 II=0
7910 FOR I=0 TO N
7920 IF FS$(I)="01" OR FS$(I)="81" THEN *TM. TO ELSE *TM. T1
7930 *TM. TO
7940 IF NT$<NS$(I) THEN *TM. T2
7950 II=II+1
7960 *TM. T1
7970 NEXT I
7980 *TM. T2
7990 TN=II
8000 IF TN>N OR I>N THEN TIME$ OFF:GOTO *MAIN
8010 '
8020 N$=NS$(I):GOSUB *DIV. NAME
8030 ON TIME$=TM$ GOSUB *TIME. T
8040 RETURN
8050 '

```

FA/MAのワープロ機能を検証する

Sec.1 FA/MAの日本語処理能力を探る

Sec.2 バックスペースキーを親指で「親指・BACKSPACE」

FA/MAのサウンド機能のすばらしさは前ページまでで十分わかりいただけたと思う。では、新しいROMが搭載されたことによって日本語処理能力はどう変わったか。ワープロとしてのFA/MAの能力と、そのあとにバックスペースキーを親指で変換可能にするプログラム「親指・BACKSPACE」を搭載する。

1 FA/MAの 日本語処理能力

改良ポイント

日本語処理能力の点で見た場合、MAのハード上の改良点は、辞書をROM化して搭載したことだけである。だが実用面から見た場合は、次に示すように実のある改良がなされている。

- ①辞書語数が4万語から6万5千語に増加
- ②辞書のROM化にともない変換速度が向上
- ③連文節変換が可能（従来は文節変換）
- ④システムディスクのフリーエリアが拡大
- ⑤日本語文書の編集ができるN88エディタをサポート
- ⑥新しいユーティリティソフトのサポート
一方FAでは、ハード的にもソフト的にも改良は皆無だ。市販ワープロソフトの文書ファイルの変換ユーティリティが新たにサポートされただけにとどまっている。

メモリアップ

従来機種やFAでは、辞書はシステムディスク上に記憶されている。このため、漢字変換時にいちいちディスクをアクセスし

なければならず、変換に多少時間がかかる。また、システムディスクの容量の大部分を辞書に占領されるおかげで、フリーエリアが少ないという問題があった。

MAは256Kバイトの容量の辞書のROMを搭載することで、これらの課題をクリアした。しかも辞書語数はMHの約40,000語に対して65,000語と大幅にアップ。そして、変換速度に余裕ができたため、より高度な連文節変換が可能となったのである。

図1と図2に、MAとFAのメモリアップを示す。MAの辞書ROMは、C000H～FFFFHのアドレスに、32バンク重ねて配置されている。

われわれユーザーが辞書ROMを直接アクセスする必要はまったくないが、システムのことは何でも知りたいという「その筋」の人のために、辞書ROMのアクセス方法を公開しよう（せっかく解析したから発表

したい）。

辞書ROMのオープンには、I/OポートF1Hに00Hを出力する。01Hを出力すれば、メインRAMにもどる。辞書ROMのバンクの選択は、I/OポートF0Hにバンク番号を出力することで行う。いうまでもなく、辞書ROMオープン時にDI状態にしなければならない。

MAは、32Kバイトの拡張RAMを4バンクもっているが、日本語BASICの日本語変換ルーチンなどが、この拡張RAM上に常駐している（システムリセット時に、システムディスクよりロードされる）。

一方、FAは拡張ROMを、日本語変換ルーチンはテキストエリアの後半16Kバイトに配置される。このため、FAではテキストエリア容量が本来の半分になってしまうのだ。これはFHのときからそうなのだが、FAでは改善してはしなかったところ

図1 PC-8801MAの場合

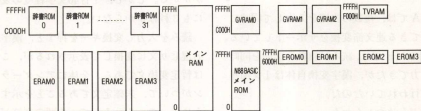
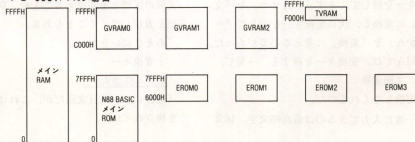


図2 PC-8801FAの場合



漢字コードについて

JIS コードの第 1 水準の全角漢字は2121 H から始まるが、シフト JIS は8140H から始まる。

N88-漢字 BASIC では、漢字の内部表現にシフト JIS コードを採用している。このコード体系だと、ASCII 文字と漢字の混在文字列を表現するのに、ほかの方式よりも少ないデータですむ特徴がある。図 3 に、各 BASIC での、

AB 日本語 CD

という文字列の内部表現を示す。

漢字と ASCII 文字を混在させるには、両者を区別するための工夫が必要となる。PC-9800 シリーズの N88-日本語 BASIC の場合は、漢字が始まるところに KI (漢字イン) コードを、終わるところに KO (漢字アウト) コードを付加している。

これに対して N88-日本語 BASIC では、特別な KI, KO コードをもち、シフト JIS 漢字と ASCII 文字がそのまま書き込まれる。普通の方法では ASCII 文字が漢字を見分けない。しかし、シフト JIS コードは8140H からなので、先頭バイトは81 H 以降。コード7FH 以下の ASCII 文字となれば区別できるのだ。その代わり、N88-日本語 BASIC では、ASCII コード80 H 以上のグラフィックキャラクタを使用することはできないようになっている。

MAの連文節変換

MA では、複数の文節を入力して同時に変換できる連文節変換をサポートしている。従来の MH では、読みは複数の文節を同時に入力できたが、漢字変換自体は 1 文節単位で行われていたのだ。

ぶんせつへんかん

という読みを入力したとき、MH では重変換キーを押して、まず「ぶんせつ」を「文節」に変換し、次に変換キーを押して「へんかん」を「変換」に変える必要があった。

MA では、変換キーを押すと、一発で

文節変換

と変換してくれるのだ。

一度に入力できるのは最高95文字、16文

図 3 各 BASIC の全角文字の表現

BASIC名(発売元)	対象機種	全角文字の表現 (例として“AB日本語CD”を16進数で表現した)																													
N88-日本語BASIC(86) (日本電気)	PC-8801MH PC-8801FH PC-8801mk IIMR/ PC-8801mk IIFR/ PC-8801mk IISR用	シフトJISコードを使う。全角文字は2バイト、半角文字は1バイトで表現される。 (例) <table><tr><td>A</td><td>B</td><td>日</td><td>本</td><td>語</td><td>C</td><td>D</td></tr><tr><td>41</td><td>42</td><td>93FA</td><td>9678</td><td>8CEA</td><td>43</td><td>44</td></tr></table> 10バイト										A	B	日	本	語	C	D	41	42	93FA	9678	8CEA	43	44						
A	B	日	本	語	C	D																									
41	42	93FA	9678	8CEA	43	44																									
N88-日本語BASIC(86) (日本電気)	PC-9800シリーズ用	JISコードの前にKI(漢字イン)、後ろにKO(漢字アウト)をつける。 (例) <table><tr><td>A</td><td>B</td><td>KI</td><td>日</td><td>本</td><td>語</td><td>KO</td><td>C</td><td>D</td></tr><tr><td>41</td><td>42</td><td>1B4B</td><td>467C</td><td>4B5C</td><td>3B6C</td><td>1B4B</td><td>43</td><td>44</td></tr></table> 14バイト										A	B	KI	日	本	語	KO	C	D	41	42	1B4B	467C	4B5C	3B6C	1B4B	43	44		
A	B	KI	日	本	語	KO	C	D																							
41	42	1B4B	467C	4B5C	3B6C	1B4B	43	44																							
8801漢字BASIC (システムソフト)	PC-8800シリーズ用	JISコードの、8ビット目と16ビット目をONにしたコードの前後に漢字イン、漢字アウトをつける。2バイトの半角文字、4分の1文字も扱うことができる。 (例) <table><tr><td>A</td><td>B</td><td>KI</td><td>日</td><td>本</td><td>語</td><td>KO</td><td>C</td><td>D</td></tr><tr><td>41</td><td>42</td><td>7F</td><td>C6FC</td><td>C6DC</td><td>B8EC</td><td>7F</td><td>43</td><td>44</td></tr></table> 12バイト										A	B	KI	日	本	語	KO	C	D	41	42	7F	C6FC	C6DC	B8EC	7F	43	44		
A	B	KI	日	本	語	KO	C	D																							
41	42	7F	C6FC	C6DC	B8EC	7F	43	44																							
(8801漢字BASICは東海クリ エイトとの協同企画商品)																															
N88-漢字BASIC (PC-8834-2W(KI)) (日本電気)	PC-8800シリーズ用	JISコードを、下位バイト、上位バイトの順に並べ、前にKIをつける。2バイトの半角ひらがなと、1バイトの半角文字の前にはそれぞれ特殊な制御文字(下の例の*の文字)をつける。 (例) <table><tr><td>*</td><td>A</td><td>B</td><td>KI</td><td>日</td><td>本</td><td>語</td><td>*</td><td>C</td><td>D</td></tr><tr><td>FD</td><td>41</td><td>42</td><td>FA</td><td>7C46</td><td>5C4B</td><td>6C38</td><td>FD</td><td>43</td><td>44</td></tr></table> 13バイト										*	A	B	KI	日	本	語	*	C	D	FD	41	42	FA	7C46	5C4B	6C38	FD	43	44
*	A	B	KI	日	本	語	*	C	D																						
FD	41	42	FA	7C46	5C4B	6C38	FD	43	44																						

節まで、数字、片仮名、句読点を含めた読みを、漢字仮名まじりの文に変換できる。したがって、次のような一発変換が可能だ。

「MAのれんぶんせつへんかんは、16ぶんせつまでをいちどにへんかんできる。」

↓変換キー

「MAの連文節変換は、16文節迄を一度に変換できる。」

読みのとおり文節を気にせず入力できるのである。辞書がROMとはいっても、長い文を一度に変換するときは、多少時間がかかる。しかし、1行40文字程度の変換にも3秒はかからない。

読みを入力し変換キーを押すと、漢字仮名混じり文に変換して表示されるが、これは暫定変換文である。全体にアンダーラインがついて、未確定文であることを示すようになっている。そして、先頭の文節がリバース表示となる。これを注目文節という。

文節の区切りを間違えることや、別の同音語を表示してくることもある。

「あそびにきたい」

↓変換キー

「遊びにきたい」

「遊びに」が注目文節だが、これは正しく変換されている。

「生きたい」は正しくない。注目文節は矢印キーで移動できるので、いまの場合□キーを押して、「生きたい」を注目文節にする。そして、変換キーを押すたびに、

活きたい
いきたい
行きたい
往きたい
去きたい
逝きたい

といった候補を出してくれるので、望みの候補を出してやる。学習機能があるので、一度選んだ候補は次からは1回目に表示される。

文節の区切り方が違っていたときは、SHIFTキーを押しながら矢印キーを押すことで、注目文節の長さを変えることができる。

たとえば、「汚職事件」と入力したいとする。初めは、

お食事券

と表示される。これは全体を1文節として判断してしまっている。この場合、シフトしながら□キーを「じけん」の読みの長さの3回を押すと、

お食事券

↓

おしょくじけん

↓
汚職事件

↓
汚職事件

というように、表示が変わっていく。注目文節の長さを変えると、それ以降の文節の表示も対応して変化するようになっているのだ。

連文節変換の効率をチェックする

連文節変換のヒット率は、かなり高くて驚いた。いろいろな文句で試してみたが、かなりよい確率で変換してくれる。

もちろん、間違えることもある。

この機械にこの機械を買おう

MAの変換機能は、文脈チェックをしないのだから、「機械」と「機会」を見分けられないのは当然だ。また、この程度の間違いは簡単に直せる。

しかし、かなりトチることもある。

この機能を使わない手はない

という文を入力したいのに、

この昨日をつ川名いて花井

となってしまった。このような、文節の切り方の間違いについては学習機能がないので、何度やっても同じように間違えてしまう。

それでもまだ、この程度なら簡単に正しく変換し直せる。次の文は、変換作業にかなりこずった

二足歩行とその制御

これは、一発目の表示は

にそくほこう屠蘇の制御

となる。これを正しく変換する操作を図4に示す。このときの操作でとくに困ったのは「そく」を変換して、変換キーでも単漢字キーでも「足」に変換してくれない。結局、ほかの変換を済ませてから最後に「あし」と入力して「足」を出さざるを得なかった。

この例などでは、変換作業をはなからあきらめて、初めから文節単位に区切って入力し直したほうが断然速い。

連文節入力の場合、片仮名変換キーが働かないのも困ったことの1つ。

風の谷のナウシカ

一度には入力できない。ただしMAの辞書は片仮名の言葉もかなりたくさんもっているため、固有名詞以外の片仮名ならば

図4 MAの連文節変換の正しい操作方法

にそくほこうとそのせいぎょ

↓ [変換]

にそくほこう屠蘇の制御

↓ [SHIFT+] × 5回 「に」に注目文節を移動

にそくほこうとそのせいぎょ

↓ [→] + [SHIFT+] × 6回 「そく」に注目文節を移動

にそくほこうとそのせいぎょ

↓ 「そく」は、自動変換でも単漢変換でも「足」に変換できない

とありあらず、「そく」はそのままだしておく

↓ [→] + [SHIFT+] × 6回 「ほこう」に注目文節を移動

にそく歩行とそのせいぎょ

↓ [→] + [SHIFT+] × 4回 「その」に注目文節を移動

にそく歩行と蘇のせいぎょ

↓ [変換] × 4回 「蘇の」を「その」にする

↓ それにともない、続く文節は自動的に「制御」となる

にそく歩行とその制御

↓ [決定]

にそく歩行とその制御

↓ [→] × 7回 未変換の「そく」にカーソル移動

↓ 「あし」と読みを入力

にそく歩行とその制御

↓ [変換]

にそく歩行とその制御

↓ 「そく」の「く」が残ってしまうので、これを削除

二足歩行とその制御

一気に入力できる。

究極のパーソナルコンピュータ

この文は、一発変換だった。

MAの連文節変換は、予想以上にヒット率が高かったが、当たらなかったときは、変換作業が非常に面倒になることがある。

連文節変換機能は、文節ごとに変換キーを打てば文節変換として使うこともできる。このように小刻みに変換キーを使うのと、ある程度の長さを入力してから変換キーを押して連文節変換を積極的に利用するのと、総合的にどちらが効率（訂正操作を含めた総ストローク数）がよいかわからないところだ。

個人的には、連文節変換は便利なので、仮に多少効率が悪かったとしても、この機能を積極的に使いたいと思う。

ちなみに私が使っているC社のワープロも連文節変換が可能だが、「この機能を…」の例文も、「二足歩行と…」も、一発で変換してくれる。とはいえ、MAはパソコンである。付属の日本語入力機能としては、十分過ぎるほどの性能といえるだろう。ワ

ープロじゃないのだから。

FAの熟語変換

FAの漢字変換は熟語単位で、MAに比べるとだいぶ見劣りする。辞書語数も、3万5千と少ない。

じゅくごたんのへんかん
と入力して、「重変換」キーの[F5]を押すと、

熟語たんのへんかん

となる。未決定の部分は、リバーズ表示である。続いて重変換キーを押すと、

熟語単位へんかん

となる。次の「の」は漢字にする必要がないので、SHIFTキーを押しながら[→]キーを押すことで、文節対象を「へんかん」に移してやる。そして重変換キーを押すと、

熟語単位の変換

となるのだ。

いまのは読みを先に入力しておき、あとから熟語単位で変換したが、入力時に熟語ごとに変換キーを押す方法も使える。この

ほうが楽だと思う。

漢字変換は、そのたびにディスクの辞書を読むが、その割にはけっこう速く、変換キーを押してから候補を表示するまで2秒とかからない。

日本語ユーティリティ

日本語関係の管理を行うための、5つのユーティリティプログラムが付属のシステムディスクに用意されている。ただし、次に示す1から4までのユーティリティは、FH、MHでサポートされていたものと基本的に同一である。

1 外字管理

外字のデザインを行い、システムディスク上に外字データの登録を行なうためのユーティリティだ。

外字作成画面は写真1。カーソルをテンキーの12346789で8方向に動かし、5でドットセット、0でリセットとなっている。画面表示用の16ドットフォントと、プリンタ用の24ドットフォントの2つをデザインできる。

画面下には、定義してある外字の一覧が表示されている。

漢字ROMのフォントを呼び出して、それに手を加えてデザインすることもできる。

作成したフォントは、システムディスクに登録、プリンタでダウンロードできる。また、登録してある外字を、別のシステムディスクに転送することもできる。

機能的にまずまずであるが、欲をいえばマウス対応にはしなかった。また、デザイン機能で、24ドットデザインを自動縮小して16ドットにする機能が欲しいところだ。

なお、JISコードの7621Hから765FHまでの63文字が外字領域となっている。ところが、N88-日本語BASICでは、漢字をシフトJISコードで扱うため、実際の使用時には外字のJISコードをシフトJISコードに変換して使う必要がある。

2 プリンタへの外字ロード

システムディスクに登録されている外字フォントを、プリンタにダウンロードするためのユーティリティ。PC-PR系プリンタの制御コードを持つ24ドットプリンタに

対応する。

なお、プリンタが接続されていないと、プリンタを準備して下さいとメッセージを出してくれるが、これは親切だ。

3 ユーザー辞書管理

辞書にない熟語を登録するためのユーティリティ。

熟語の登録、登録してある熟語の変更、削除、ユーザー辞書の一覧表示（プリンタ出力可）、ユーザー辞書のほかのディスクへの転送などの機能がある。

熟語は全角文字で16文字以内でなければならない。

熟語の登録は、文書作成中必要を感じた時点で行いたいものだが、残念ながらそれではできない。これにはその都度ユーティリティを起動しなければならない。

4 コード変換

N88-日本語BASICではシフトJISコードを採用しているが、PC-9800シリーズのN88-日本語BASIC(86)をはじめ、ほかのBASICでは、別の漢字コードを採用している。これらの中でプログラムやデータの交換を行うためには、漢字コードの変換が必要だが、それを行うためのユーティリティも用意されている。

交換できるBASICの種類は、次の3つ

- ① N88-日本語BASIC(86)
- ② N88-漢字BASIC
- ③ 8801漢字BASIC(システムソフト)

ファイル形式は、BASICプログラム、ランダムファイル、シーケンシャルファイル。ドライブのタイプは、FAでは2Dのみだが、MAでは2D、2DD、2HDの間で相互に転送可能である。

5 文書ファイルの変換

Super春望、JET8801A、ユーカラK2(art)の3つのワープロ文書と、N88-日本語BASICの間で、相互にファイル変換を行う。これはFA、MAで新たにサポートされたユーティリティである。

N88エディタとは?

MAの場合、N88エディタという日本

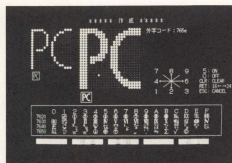


写真1 外字作成画面

語の文章ファイルを編集できるユーティリティが付属している。

N88エディタを使う場合は、ファイル数を1にしてN88-日本語BASICを起動し、
RUN "n88ed. ip 1" ②

とする。

漢字の入力に関しては、N88-日本語BASICと同じ連文節返換が使える。ただし、入力中の文は黄色で、注目文節は水色で表示される。

入力中は、常にインサートモードになっていて、上書き（オーバーライト）はできない。したがって、タイプミスは、訂正文字を削除してから正しい文字を入力するか、逆に訂正文字を挿入してから不要な文字を削除することになる。

X68000は良い!

↓ Xにカーソルを合わせ、PC-88

VAと入力

PC-88VAX68000は良い!

↓ 6回削除キーを押す

PC-88VAは良い!

常時インサートモードは、初めは面倒に感じるが、慣れると誤訂正が少なく、むしろ使いやすい。

単語のサーチ、置換も可能。文章の部分コピーや移動など、一応日本語ワープロとして使える機能を有している。ただし、文章の幅は画面上では全角40文字に固定で、変えることはできない。

おもしろいことに、コントロールコードまで文書中に含まれることができる。場合によっては価値のある機能だ。

単語の登録も、文書の作成中に可能。

COPYキーを押すと、編集中の文書をプリンタに出力する。残念ながら、部分的な印刷はできない。また、文書中のコントロールコードは、すべて(CRとLFはのぞく)スペースとして出力される。

完成した文書は、ディスクにファイルとして保存することになるが、2D、2HDのデ

F337 3AF9EF
F33A FE03
F33C 2820

LD A, (0EFF9H) — キースキャンフラグを読む
CP 3 — 3ならば、入力なしなので、
JR Z, POPRET — 以下の処理はしない

F33E DBE2
F340 2F
F341 F5
F342 AF
F343 D3E2

IN A, (0E2H)
CPL
PUSH AF
XOR A
OUT (0E2H), A — 拡張RAMをマスクする

F345 3AC2E6
F348 F5
F349 E6F9
F34B D331

LD A, (0E6C2H)
PUSH AF
AND 0F9H
OUT (31H), A — ROM, RAMモードにする

F34D 3AFEEF
F350 FE01
F352 2818
F354 FE20
F356 280C

LD A, (ASCIN) — 入力したキーのASCIIコードを読む
CP 1 — 全角モードのとき、決定キーのコードは1
JR Z, KBACK — 決定キーが押されたのなら、KBACKへ
CP ' ' — 半角モードでは、決定キーはスペースとして働く
JR Z, ABACK — その場合はABACKへ

F358 F1
F359 D331
F35B F1
F35C D3E2

CLOSE: POP AF
OUT (31H), A — メモリモードをもとにもどす
POP AF
OUT (0E2H), A — 拡張RAMの状態をもとにもどす

F35E E1
F35F D1
F360 C1
F361 F1
F362 FB
F363 C9

POPRET: POP HL
POP DE
POP BC
POP AF
EI
RET — 終わり

F364 DB0D
F366 CB4F
F368 280C
F36A 18EC

ABACK: IN A, (0DH)
BIT 1, A — 決定キーが押されているかチェック
JR Z, BACK — 押されていればBACKへ
CLOSE — そうでなければCLOSEへ

F36C DB08
F36E CB77
F370 28E6

KBACK: IN A, (8)
BIT 6, A — SHIFTキーが押されているかチェック
JR Z, CLOSE — 押されていればCLOSEへ

F372 AF
F373 CD1B46
F376 AF
F377 CD1B46

BACK: XOR A
CALL 461BH — キューバッファから1文字削除
XOR A
CALL 461BH — キューバッファから1文字削除
— 全角モードの決定キーは、
2バイトのコードで表されるため、
キューからは2バイト削除する必要がある

F37A AF
F37B 1E08
F37D CDD45

XOR A
LD E, 8
CALL 45DDH — キューバッファに
バックスペースコード
をセットする

F380 AF
F381 32FEEF

XOR A
LD (ASCIN), A — リピートキャンセルのための処理

F384 18D2

JR CLOSE — CLOSEへ

F386

END

PC-88VAのUp Grade Board ビデオアートボードを活用する

PC-88VAのビデオアートボードについては、「Oh! 88」1号ですでにご紹介しましたが、本誌ではその具体的な活用例についていくつかの作例を上げて解説します。なお、このページで紹介した作例の画像をパッキングしたソフトウェアをスタジオBCから本誌の読者にプレゼントします。詳しくは158ページをご覧ください。

スタジオBC

石葉直志

VA・その後

PC-88VAが発売されてから約1年の間、その性能については、いろいろなところで取り上げられました。

最大の特徴65,536色同時表示可能なグラフィックス、ビデオ出力端子を標準装備など、VAに少しでも興味をもっている人ならすでにご存じのことでしょう。ハードの性能はいうまでもなく非常に高度なものです。ソフトウェアの不足を嘆いていた当初に比べ、NECの「アニメフレイマー」やツァイトの「Z's STAFF Kid VA」など質の高いものが揃いつつあります。これらはVAの性能を生かすために多機能で、使いこなすには時間を要しますが、慣れると実にいろいろなことが可能になってきます。

素晴らしい可能性と機能を秘めているVA。これからはソフトウェアの分野でそれを「どう生かすか」にかかっているのではないのでしょうか？

ビデオボード・「PC-88VA11」

今回はVAのオプション、ビデオボード「PC-88VA11」(写真A)の活用法について解説したいと思います。

このビデオボードはVA本体の専用スロットに実装することによって、外部のビデオ信号のデジタル化、スーパーインポー

ズを可能にするVA専用のボードのことです。

VAをお持ちの方、これから購入しようという方でビデオを持っているなら、VAを十分に活用するためにもいいツールとなることは絶対です。

なお、ビデオボードの接続の方法は本誌の前号で解説してあります。

ビデオボードの機能

では、ビデオボードでなにができるのでしょうか？

それは大きく分けて以下の2つがあげられます。

- ①ビデオデジタル化機能
- ②スーパーインポーズ機能

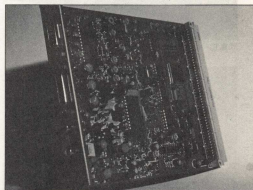
①はビデオの画像をデジタルな静止画像として取り込み、VAで画像の再現を可能にします。

②はビデオの画像にVAで作った文字や絵を重ねて表示できる機能です。

では、これからこの2つについていくつかの活用例をあげながら解説していきましょう。

ビデオデジタル化機能

VAを実際に使うとよくわかりますが、プログラムを組むにしても、グラフィックエディタで絵を描くにしても、65,536色と



写真A ビデオボード「PC-88VA11」

いう膨大な色数を使いこなすのは大変なことです。従来のパソコンと同じ感覚ではとてもついていけません。もっとも、全色無理に使う必要はありませんが、せっかくある機能を遊ばせるのはもったいないのではないのでしょうか？

このグラフィック機能の本領を発揮させるのがビデオデジタル化機能です。

従来の16色、256色、512色のデジタル化とは比較にならないほど美しく鮮明な画像が取り込めます。ビデオをスチル画面に取り込んだものとはほとんど差がなく、まるで写真のようなリアルな画像が得られます。

そしてこの写真のような絵をグラフィックエディタやそのほかのプログラムで自由に加工できます。

これがビデオデジタル化のおもしろいところです。

その具体例を3つ作ってみました。

なお、今回はツァイトの「Z's STAFF Kid VA」を使用しました。



写真B 変身前



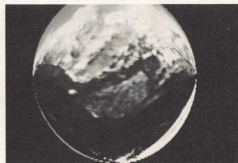
写真C 変身中



写真D 変身後



写真E もとのデジタル画像



写真F Eをプログラムによって丸めたもの



写真G Eをプログラムによってゆがめたもの



写真H 左側メガネ用



写真I 右側メガネ用



写真J 赤青メガネ用3D静止画像

Exam.1 ソフトウェアによる作画 友達の顔を変えちゃおう!

HOW TO MAKE

- ①まず、ビデオボードをセッティングしてから、ソフトウェアを立ち上げデジタルイズのコマンドを選びます
- ②ビデオを再生して、欲しい絵が出てきたらデジタルイズします。このソフトの場合はオプションのビデオボードを選び、再生してからマウスのボタンをクリックします
- ③今度は加工です。まずバルカン人の特徴となる耳や眉を描き、目の上と鼻から下を長くします。これだけだと継ぎ目が目立つので、似たような色合いのところを探してコピーしていきます

以上のことを時間をかけて作っていけば、写真⑩のようなリアルなものが作れます。

また、このように変形したものの途中の段階を1つずつビデオに落としていけば、SFX映画を作ることだって可能です。

あなたもいろいろ挑戦してみてください。

Exam.2 プログラムによる作画 雪山絶景3

HOW TO MAKE

今度は自作のプログラムによって絵を加工作する方法です。

写真⑬⑭⑮は、元の絵である⑬を BASIC のプログラムによって丸くしたり(写真⑭)、シンカーブにそってゆがめたり(写真⑮)したものです。

このようにコンピュータのプログラムで加工すると、よりリアリティな凝った画像を作ることができます。

これらの絵も「Z's STAFF Kid VA」によってデジタルイズしているので、BASICで

絵を読み込み際に、付属の「ZLOAD.BIN」というプログラムを使用しています。どれも簡単な BASIC のプログラムですので、いろいろな応用ができると思います。

Exam.3 ビデオカメラを使用する 3Dグラフィックス

HOW TO MAKE

近頃、3Dという言葉もめずらしくなくなり、家庭用のゲームに3Dが登場し、ビデオカメラにも3D用の特殊なものが登場してきました。

初めて3Dを見たのは筑波科学万博でしたが、こんなに早く普及するとは思っていませんでした。

家庭用ゲームやビデオのものは、液晶切り替えによるカラーの3Dで、専用のメガネが必要になります。そこで今回は、普通のメガネの両側に赤と青のセロハンを貼

て自作の専用メガネを作りました。

では、メガネもできたことですので、作例の説明に入ります。

①ビデオボードをセッティングし、ビデオカメラをボードのビデオ入力端子につなぎ、ソフトウェアを立ち上げます

②被写体をセッティングし、図1のように2つの方向から片方ずつデジタイズし、それぞれディスクにセーブします。このとき、カメラの高さが変わらないように注意します（三脚があると便利）

③BASICを起動した後、プログラム（リスト1）をRUNします。Exam. 2と同様のファイルを使用します

④次にメガネの左側用のための絵を読み込み、色指定の際「1」を入力します

⑤画像をディスクにセーブします。今回のソフトの場合、1ドライブにシステムディスクを入れ、ESCキーを押しながらRESETを押してセーブします

⑥④と同様に、今度はもう一方の絵を読み込み、色指定の際は「2」を入力します。右側用の絵を作り、ディスクにセーブします

⑦そしてこの2枚の絵を1枚に重ね合わせればできあがりです

この重ね合わせの方法には2とおりあります。

1つは、2枚の絵の同じ座標のドットを照らし合わせて、その変わった色を画面上に表示していき、1画面分を自動的に作成していくプログラムを組む方法。

もう1つは、タイリングを使う方法。

今回はソフトウェアの関係上、後者の方法ですべての作業ができてしまうので、この方法を採用しました。

以下にその手順を説明します。

⑧ソフトウェアを起動し、図2と図3のようにトーンを作ります

⑨右側用の絵を読み込んで、0番の絵を選び、図2のトーンを画面全体にかけます

⑩メニューのアイコン（図3A）を使って、画面を50ドット下にずらします。そして絵の上半分をその上にコピーします。同様にアイコン（図3B）を使って絵の下半分をその下にコピーします

⑪画面を元の位置にもどし、左側用の絵を読み込んで0番の色を選び、図3のトーンを全体にかけます

図1

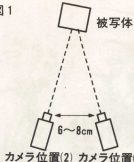
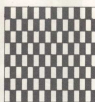
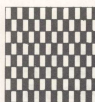


図2



□...白(67535)

図3



■...黒(0)

図3A



図3B



写真⑧ 新宿高層ビル



写真⑨ 左右2つのビルを消したものと

⑫0番を除いたコピー（詳しくはソフトのマニュアル22ページ）を使って、⑬の逆の手順で上下半分ずつになった右側用の絵を片方ずつ左側用の絵に重ねます。そして画面を元の位置にもどしてやれば完成です

「Z's STAFF Kid VA」を持っていない方にはわかりにくいかもしれませんが、赤と青の絵を市松に重ね合わせているだけのことです。メガネを作るとき注意としては、左側用の青い絵が、黒く見える程度の赤色のセロハンを右側にもつけてます。そして、右側用の赤い絵が、黒く見える程度の青いセロハンを左側にもつけてます。

作例はVAを使用していますが、静止しているものなら何でも大丈夫です。

ぜひいろんな工夫をして3D画像を作ってみてください。

その他の活用法

これまでに紹介した3つの例のほかに、現実をシミュレートさせて活用する方法があります。

具体的に、ビルを取り壊したときにその街の風景はどう変わるかを、ビデオ上で想定して見ることが出来ます（写真⑩⑪）。あるいは、その逆で、空き地にビルを建てた場合のシミュレートもすることが出来ます。

こういった、映像の上でのシミュレーションがビデオボードによって可能になります。たとえば、これをパーソナルユースで

部屋の模様替えなどに役立てることも可能です。

本誌のカラーページに掲載されている沢井さんの電話機も、模様をデジタイズしてVA上でカラーパリエーションをシミュレートさせています。

このように、ビデオデジタイズだけでもVAの活用範囲がこんなに広がるのです。

スーパーインポーズ機能

ビデオボードを使うことによって可能になるもう1つの機能はスーパーインポーズと呼ばれる機能です。

これは、VAで作ったグラフィックスと、外部から入力したビデオの画面を重ねて表示させる機能です。

たとえば、自分で撮影したビデオの冒頭にタイトルの文字や絵をつけたり、ビデオの動く背景の上にアニメを動かしたりすることが可能になります。

従来のパソコンでもスーパーインポーズ機能を持ったものがありましたが、ビデオにテロップをつける程度のもものがほとんどで、「ビデオのためにパソコンを利用するもの」がほとんどでした。

ところが、VAには「アニメフレイマー」というソフトウェアがありますので、そのスーパーインポーズ機能を使ってアニメとビデオの合成を楽しむことも可能なのです。

では、「アニメフレイマー」を使ったグラフィックスの作例を家庭用ビデオで作る例とコマーシャルで使われた例の2つをご紹介します。

紹介します。

Exam. 4 ビデオにアニメで タイトルをつける

HOW TO MAKE

家庭用ビデオが廉価になったおかげで、旅の記録や子供の成長記録に利用している人が増えたようです。

これはそういったものを編集する際に、アニメでタイトルをつける楽しみ方です。

作例(写真⑧)は、友人のスキー旅行のビデオに、タイトルをアニメーションでつけたものです。これには、再生用と録画用ビデオ2台が必要になります。これなら、いろんなビデオに思い思いのアニメーションを組み合わせて楽しむことができます。

新製品のデモンストレーションで登場したVAのグラフィックス

HOW TO MAKE

ここでは、いままでの例とは少し外れますが、VAのグラフィックスがいかに並外れたものであるかを証明する事例をご紹介しますと思います。

先日、ある大手化粧品メーカーが新しい口紅を発表しました。そのプロモーションビデオの一部にVAで作成した画像が使われています(写真⑨⑩⑪)。

これは、「Z's STAFF Kid VA」で作ったグラフィックスをビデオ出力端子を使って録画し、それに「アニメフレイマー」を使ってアニメをスーパーインポーズしたものです(編集ではVAではなく、ほかの機械が使われています)。

いままで、テレビやビデオで使われていたコンピュータグラフィックスは、高価な、パーソナルユースとはほど遠いコンピュータが使用されていました。

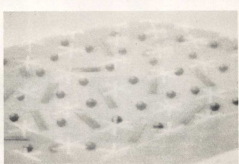
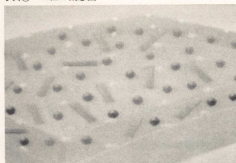
しかし、この事例を見てもわかるように、今ではパーソナルユースのパソコンで、テレビでも使用できる質の高い画像が得られるようになったのです。

VAは、これまでのパソコンとはひと味もふた味も違う、まさに新世代のパーソナルコンピュータなのです。

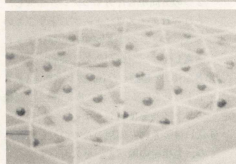


写真⑧ スキーのビデオに、タイトルアニメーションをつけた例

写真⑨ 口紅の概念図



写真⑩ ネットが少しずつ広がっていきます



写真⑪ 完成図

```
1000 ' save "dw2.bas",A
1010 '
1020 ' O h 8 8
1030 '
1040 ' Z' s s T A F F -> セノクロ 64 階層
1050 '
1060 ' St.B.C T.Kimura 1988.1
1070 '
1080 '
1090 '
1100 SCREEN WIDTH 4,640,16
1110 CONSOLE 0,25,0,0
1120 WIDTH 80,25
1130 SCREEN ,2
1140 CLS 1
1150 OUT #H100,98
1160 CLEAR #H8000
1170 DEFINT A-Z
1180 DIM BF(641)
1190 '
1200 GOSUB *ZLOAD
1210 '
1220 *MAIN
1230 '
1240 GOSUB *S.MCODE
1250 COL=0: SCREEN ,2
1260 WHILE (COL<1 OR COL>2)
1270 INPUT "色を入力してください。 青=1 赤=2 : ",COL
1280 WEND
1290 COL=COL-1: CLS 1: SCREEN ,0
1300 '
1310 FOR Y=0 TO 199
1320 GET (0,Y)-(639,Y),BF(0)
1330 FOR X=2 TO 641
1340 CALL BF(BF(X))
1350 NEXT
1360 PUT (0,Y),BF(0),PSET.
1370 NEXT
1380 '
1390 BF(0)=1: BF(1)=1: BF(2)=COL*961+31
1400 PUT (0,0)-(639,199),BF(0),AND
1410 '
1420 PRINT "AドライブにZ' s s T A F F のシステムディスクを入れ"
1430 PRINT "エスケープキーを押しながらリセットしてください。"
1440 END
1450 '-----
1460 '
1470 *ZLOAD
1480 '
1490 DEF SEG=#H8000
1500 BLOAD "zload.bin",0
1510 DEF USR=#A80
1520 FILES "b:*.zim"
1530 INPUT "ファイル名を入れてください。",NS
1540 NS="b: "+NS+".zim"
1550 CLS 1: SCREEN ,0
1560 '
1570 NS=USR0(NS)
1580 '
1590 RETURN
1600 '
1610 '
1620 *S.MCODE
1630 DEF SEG=#H8000: BW=0: AD=0
1640 RESTORE *M.CODE: READ DS
1650 '
1660 WHILE (DS<>"**")
1670 POK DS,VAL("H"+DS):AD=AD+1
1680 READ DS
1690 WEND
1700 '
1710 RETURN
1720 '
1730 *M.CODE
1740 DATA 89,85,06,C4,76,04,26,8B,04,BB,1F,00,21,C3,B9,E0
1750 DATA 03,21,C1,C1,B9,05,C1,E8,0A,01,D8,01,C8,D1,E8,89
1760 DATA C3,C1,E3,0A,01,E8,09,C3,C1,E0,05,09,D8,26,89,94
1770 DATA 07,C4,04,00,90,**
```


究極の超初心者のために

最後の マ・シ・ン・語

—考え方を考えよう—

田奈真実

読者の方々からの絶大なるご要望により、
「これでわかんなきや、どーしようもない」超
初心者向けのマシン語講座を行います。Z-80
のこと、ビットのこと、それらのことをまっ
たく知らない方たちのために、今回はまず2
進法の考え方、概念から勉強します。あせら
ず、急がず、確実にステップを踏んでいつて
ください。今まで中途半端にしかわからなか
った人も、必ずわかるようになります。

● これでわかんなきや、
あきらめよう!?

さてえ、みなさん、マシン語という言葉、
聞いたことありますか？ ええっ？
TVの前のお友だちい、もっと元気な声で
いってごらん、て、ぼくはノッポさんじ
やないってば。この話はね、マシン語って
のは聞いたことあるけど、どういふものな
のかわかんないっつー人のためのお話。

「Z80のインストラクションで相対ジャン
プと絶対ジャンプだと、マシンサイクルは
どっちがどのくらい速いの？」なんて質問
をできる人ははっきりいって、これ読んだ
ら頭クラクラしちやいます。とにかく、な
んにもわからないあなたのために、超下級
究極のマシン語講座をこれから行います。

● まず、デジタルだ

まず、デジタルの話です。コンピュータ
は0と1で動いているというのは、聞くこと
が多いでしょう。デジタルというのは「数
字の」というのが元々の意味ですから、別
に0から5で動いていても、-1000から198
で動いていてもデジタルには違いないんで
すが。それでは、なぜ0と1で動いている
のでしょうか。

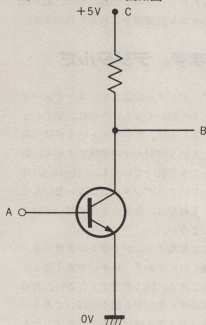
ここで登場するのがトランジスタです。
少し難しいですか？ いやいやそんなこと
はないですよ。図1を見てください。あの
丸い円の中に矢印やら線やら描いてあるの
がトランジスタと呼ばれる部品です。なん
だか技術家庭の授業のようですね。この図
1でC点（こうしただけで、理科や数学の
嫌いな人はさじを投げてしまいそうて、書
いている僕は怖いのです）には常に5ボル
トの電気を入れておきます。ここが回路の
電源になるわけです。そして、A点に電気
をかけてみます。（乾電池でかけるとすれば
1.5ボルトになります）。するとB点は5ボ
ルトになって、つまり電気が出てきたとい
うわけですね。A点になんにもつながらな
いとうなるでしょう。このときB点には電
気が出てきません。

A点につないでいた電気を弱くする、1.5
ボルトだったのを1ボルトにする、とどう
なるでしょう。やっぱりB点には電気が出
てきて、5ボルトになるんです。A点をも

つと弱くして、0.5、ええい、0.4、0.3、としていくと、この辺でB点は電気が急に弱くなって、突然0になって電気が出なくなります。ということは、すなわち入口に少しだけ電気を入れるとすぐ出口は5ボルトになって、それを少し弱くしただけで、出口には電気がなくなってしまうということになります。

コンピュータは全部が全部、このトランジスタという部品が集まってできています。つまりトランジスタのこの性質を利用して、電気を電圧があるかないかで、すなわち0と1に対応させ、これに従って動いているのです。

図1 トランジスタの回路図



電気信号の玄関・ゲート

門のことをゲートといいます。ここでは電気信号の門にあたるわけですので、図2にそのもっとも基本的なものを示します。こんなものばかりでは親しみがわかないでしょうか。ま、ここもひとつくらえて先へと進んでください。

このゲートはNAND（ナンド）と呼ばれます。AとBとどちらにも電気が入ったとき、ということつまり、どちらも1になったときに、Cが0になります。さて、これがなんになるのでしょうか？

入口がどちらも1のときに出口は0。これを組み合わせるといろんなことができて、

たとえば、2進数の足し算なんていうのは簡単にできます。さらに組み合わせると、入口の1つに1を入れておくで出口には足し算の結果が出てくる、というのもできます。

さらにさらに組み合わせると、入口Aが1で入口Bが0のときに、足し算の結果が出口に現れ、入口Bが1なら引き算の結果が出てくる、なんていうのも当然可能なのです。

さらにさらに、これを組み合わせることによってデータって記憶しておける、つまりメモリを構成することもできるのです。

コンピュータの中身は、さっきいったトランジスタの組み合わせです。トランジスタの組み合わせはゲートを作り、これが集まってコンピュータになります。このような部分が集まっているわけですから、コンピュータの内側では何もかも全部、0と1の組み合わせで表現されます。そこは2進数の世界なのです。

では、2進数とは

マシン語が出てくるときに必ず2進数という言葉が聞かれるのは、このように0と1だけの世界で数表現するにはこの方法しかないためなのです。では、2進数とはどういうものなのでしょうか。

僕たちが使っている10進数では数字が0から9の10とありあります。0よりも1が大きく、1よりも2が大きく、という数が10個あるんですね。ところが、2進数の世界には数字は2つしかない。0と1。1は0よりも大きい。これだけしかありません。

3つ、という数を出すのはどうすればいいでしょう。僕たちは「3」と聞けばわかります。ところが2進数の世界には3という数字がありません。

ところで、10という数については、それに相当する文字を僕たちは知りません。1の次に0を書き並べているわけですが、これは、9の次に大きい数を出すのに文字がありませんから、2つの数字を組み合わせたら、そういうものなのです。9の次は一番小さい0にもどして、1回まわったよというしるしの上に、1回を意味する1をつけたものなのです。

これと同じ要領で、2進数の世界で3と

図2 ゲートの概念図



入口 → 出口

A	B	出口
0	0	1
0	1	1
1	0	1
1	1	0

いう数を出すことができます。2進数の世界でも、0は0、次は1。で、その次を10とするのです。9の次に10がきたのとまったく同じ理由です。2進数の世界の10は、1の次に大きい数ですから僕たちの「2」に当たります。

すると次に大きいのは11となって、ですから僕ら10進数の3は、2進数の世界では11と表しているのです。

2進数は1桁に0と1の2つの数字しか存在しません。僕たちは1桁で10個もの数字を使っているのに、です。1桁で表すことのできる数が少ないために、同じ数を10進数と2進数で書くと、2進数のほうが桁をやたらに多く使います。表1にその様子を示しました。2進数って思いつきり長いでしょう。これは数学の問題ですが、桁数は10進数に比べ、約3倍になります。

そこで便利なのが16進数なのです。2進数から10進数とか、その逆なんかは計算しなければ求められません。なかにはそんなの簡単だよ、って人もいるでしょう。すごいでしょね。

2進数と16進数の関係は？

だからなんなのでしょう。話を途切れさせるようなことを書いてはいけませんってば。さて、話をもどしてつと。

ところが、2進数はパッと16進数に変換することができ、16進数はパッと2進

表1 10進数, 2進数, 16進数の対応

10進数	2進数	16進数
0	→0	→0
1	→1	→1
2	→10	→2
10	→1010	→A
15	→1111	→F
16	→10000	→10
100	→1100100	→64
1000	→1111101000	→3E8

表2 10進数, 2進数, 16進数の16段階の対応表

10進数	2進数	16進数
0	→0	→0
1	→1	→1
2	→10	→2
3	→11	→3
4	→100	→4
5	→101	→5
6	→110	→6
7	→111	→7
8	→1000	→8
9	→1001	→9
10	→1010	→A
11	→1011	→B
12	→1100	→C
13	→1101	→D
14	→1110	→E
15	→1111	→F
16	→10000	→10
15+1	5ケタ	F+1

数に変換できます。

これは、2進数4桁で16進数になりますから、4桁ごとに機械的に直していけばよいのです。表1で10進数の1000が2進数の1111101000になるのは、全然わかりませんが、この2進数が「3E8」という16進数になるのは簡単にわかります。とはいうものの、ある程度の慣れは必要ですが。

つまり2進数の0から4桁までは10進数で16段階ありますから、文字が16個必要になります。で、0から9までの10個の数字とその上にAからFまでの6つの文字を当てて、それを数字として利用しているのです(表2参照)。

数が命令になる

さっき、ゲートのところで説明しましたが、コンピュータの中ではぜんぶ1と0の組み合わせで動いています。ですから、どれをとっても2進数でいい表示こ

とができます。逆にいえば、2進数に直せないことはコンピュータが処理できないものなのです。少なくとも今の段階ではそうです。

「人工知能」という学問があって、そこでは知識や感情などをなんとか2進数に直そうと日々努力しています。「感情なんか簡単に数字で割り切れるもんじゃないよ」なんていう人も結構多いんですが、事実これは当たっているようで、大変にみな苦労しています。こういったものが数字で表せる、というのは理料系である僕にしてみれば疑いはないのですが。でも「簡単に」はできそうもないのです。感情や知識を2進数にすると異常に大きくなってしまい、今あるコンピュータやメモリには簡単におさまらないらしい、というのがわかってきています。

おっと、また余談をしてしまいました。そうそう、とにかく2進数ですべて表現しているということは、マシン語では命令

図3 数を命令にした計算

$$\begin{array}{cccc} 1 & 1 & 0 & 1 \\ \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ 11 & + & 1 & = \end{array}$$

自体も2進数で表すという話なのです。

1+1の計算は?

命令が「数」であるときには計算はどうなるのでしょうか。1+1の計算であれば、

「1と1をたしてその結果を教えてね」
私たちの言葉ではこうなります。電卓では、

$$1 + 1 =$$

としますね。命令を数で表すとすればどうなるでしょうか。ここでは10進数で書いていきます。まだ、2進数には慣れていないでしょうから。2進数も10進数も数であるということで、本質的には変わらないですからね。さて、「足せ」を「1」、「結果を表示せよ」を「2」で表すとすれば、

$$1 \quad 1 \quad 1 \quad 2$$

となりますね。これを見て「イッセンヒヤクジュウニ」と読んではいけません。「イチイチニ」とこころは読んでください。どうやらうまくできたようです。さて、この調子で11+1ををやってみましょう。

$$1 \quad 1 \quad 1 \quad 1 \quad 2$$

さて、これどう読んだらよいのでしょうか。1+1=でしようか、1+1+2とも読めてしまいますね。これはまずい。数と命令がごちゃごちゃになってしまいました。

さて、どうしましょう。

まず、全部2桁で表しちゃえ、という解決法もあります。すなわち、図3を見てください。

これなら数と命令を混乱することなく読むことができます。

しかししかし、この方法では3桁の数を扱うのに命令を3桁にしなければならず、4桁なら4桁にする必要があります。

ええい、この調子で続いたら命令にムダに桁数が増えすぎてしまてはありませんか。というわけで何とかよい方法はないでしょうか。

素性を示す数「ID」

ここで、登場するのがIDです。コンピュータ用語ではIDは「識別子」ということになります。僕らにとっての識別子は免許証、社員証、学生証などですね。で、このIDをうまく使えば、さきほどの問題をうまく切り抜けることができます。

またまた、電卓の場合を考えます。

1 1 +
と押せば11が入り、次に入れる数と加算されることになります。3桁だろうが4桁だろうがこの「+」を打ったことでそれまでに入れた数字の並びが数であることを示すのです。つまり、この「+」が「足せ」という命令をして、数はここまでであるというID、識別子の役割を果たしているのです。

さてさて、数で命令を表さなければいけないということであれば、「+」が数になってしまうから電卓と同じようにはいけません。そこで、数の頭には必ず次の何桁かが数値であるのかを示すIDをつけることにします。ここで、3を1桁、4を2桁、5を3桁の数値を示すものとしします。このとき、11+1の計算は、図4のように表すことができます。ね、なかなかナイスな方法でしょう。

ジャンプを考える

さて、ここまでで電卓のような計算はなんとなく数の命令でもできるような気がしてきました。さらにここで繰り返しを考えてみましょう。コンピュータで計算や処理の手順を記憶しておき、それを繰り返すことによってさまざまな処理ができるわけで、このことが大変に重要なのです。

「人間が入力した数を2倍にして表示し、それを繰り返す」という問題を考えてみましょう。で、ここまで新たな命令を増やしましょうか。「人間に数を入れてもらう」のを6、かけ算は7、「計算して表示するけど終了しない」を8、「最初にもどる」を9としてみます。このとき、この問題は、図5のようになります。これは簡単にできました。さて、次には2という数を次々に2倍にしていって、つまり、2、4、8と表示させていく場合を考えてみましょう。

3 2 7 3 2 8 ?

おやおや、今まで作ってきた命令ではどうやらだめなようです。最初にもどらずに特定の場所にもどる必要が出てきちゃいました。

アドレスとデータ

アドレスって、あの、アドレス帳のアドレスのことです。ですから僕たちはふだん住所の意味で使っています。いま問題となっているのは特定の場所にもどれない、という点です。コンピュータでは場所を特定するのにメモリ中の住所である、「アドレス」を使います。メモリ、つまり記憶場所には1つひとつに番号が0から順につけられていて、場所を示すのにこれを用いるのです。もちろん、これも数です。

今度は0をジャンプの命令としましょう。0に続く2つの数をアドレスとして、そこにジャンプします。すると、

アドレス 0 1 2 3 4 5 6 7 8
データ 3 2 7 3 2 8 0 0 2

ね、アドレスは番地をつけて呼びますから、0番地にIDで、まず数が2、2番地はかけ算、3番地と4番地がまた2、5番地が表示、6、7、8番地で2番地へのジャンプ、でかけ算を繰り返し、その数を表示します。

さて、そこで問題です

マシン語の原理、ということであれば以上のとおり。命令もデータもすべてが数。数というより1と0の組み合わせによるコードあるいは符号というのが適当かもしれません。ここまで見てきたように、符号ですべて表現されているというのは応々にして問題もあります。とはいっても実際そうやって動いているコンピュータはたくさんあるわけですから無理なことではありません。

さて、いままで適当に作ってきた命令にしたがって次のプログラムを見てください。

アドレス 0 1 2 3 4 5
データ 4 3 0 2 3 1 2

これ、0、1、2番地で30が入り、3番地が足し算、4、5番地で1となり、6で結果を表示して、つまり30+1=で、31が得られますね。ところがどっこい、これを

図4 IDを使った計算

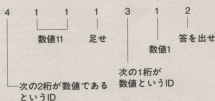
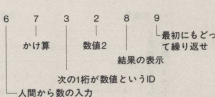


図5 ジャンプ命令の概念



1番地から読み始めてしまうと、0+1=と読むことができて、まったく違った結果が出てくるのです。

どの番地から読み始めるかが違うだけで、コンピュータにしてみれば何の誤りも検出しないでしょう。さらにさらに、このプログラム、2番地から実行を始めてしまうとどうなるでしょう。2番地は0ですからジャンプ命令ですね。その後、2、3と続き、だから23番地へとジャンプしてしまうのです。さて、23番地といってもこの状態はどうなっているか全然わからず、それが数値なのやら命令なのやらという、この状態があつた「暴走」です。プログラムに入力のミスがあつたときも同じように困ったことになります。3番地を2とすると、間違って0を入れてしまえば、31番地にジャンプして、これも暴走してしまいます。

このようにマシン語ですべてが数で、つまり符号でしかありませんから、このように恐ろしい暴走の浮き目にあることが多く、これがプログラムを組む際、大きな問題となるのです。

マシン語によるメリット／デメリット

さて、ここまで原理を書きましたが、今度はもう少し具体的な話に移りましょう。マシン語でプログラムを書くのはどうしてでしょうか。

速い/

これは皆さん知っていますよね。マシン語は回路によって直接実行されます。頭のはうで述べた NAND ゲートは、入力があったから出力まで10から20ns (ナノセカン)しかかかりません。だいたい1億分の1秒といったところです。このような回路によって直接解析されるのでから、1つの命令は100万分の1秒程度で実行されるのです。これに対して BASIC などではこの命令を1000回なり10,000回なり繰り返すことで、実行を行っているのですから、どうしても遅くなってしまいます。

細かい命令で速い操作

マシン語でなければいけない周辺機器などがあります。ゆっくりとした装置であれば BASIC の OUT 命令や POKE 命令などで十分ことが足りるでしょうけれども、多くの場合、とくに周辺装置を自分で作ったときなどマシン語の速さでコントロールしてやらないとまずいです。このような場合、マシン語でしか書けない高速なコントロールを必要とする部分のみマシン語を使い、このルーチンを BASIC なり C なり的高级言語から使うのが一般的です。このようなルーチンをデバイスドライバなんぞといいます。

以上が大きなメリットです。

プログラミングが面倒

次にデメリット、短所ですが、回路が直接実行できるような、非常に原始的な、稚拙な命令しかマシン語にはありません。複雑なことをするためには、それら単純な命令を複雑に組み合わせなければなりません。はっきりいってたいへんです。さっき、原理の話で作った命令のうち、人間からの入力を持つ、などという命令は存在しません。任意の数の足し算もありません。1命令でできるのは PC-8800 シリーズで多く使われている部品 (CPU) の場合、2進数で16桁までです。つまり、結果が10進数の65535を超えるような足し算は、それ用にやはりプログラムを組まねばならないのです。また、かけ算の命令はありません。

しかし、こういった処理はよく利用され

るために1回サブルーチンとして組んでしまえば、それを利用することはいちいちプログラムし直す必要はなくなりますが。

デバッグが大変

前にもしつこく書きましたが、マシン語は暴走します。また、それを防ぐようなデバッグ用のプログラムやら、暴走をなるべく起こさないような工夫やら、いろいろあるにはあるんですが、結局は暴走します。

わかりにくい

一度感じをつかんでしまえば、どのマシンでも似たようなものなので苦労はないのですが、とにかく単純な命令しかないために慣れるまでは、覚えづらく、使いにくいのも事実です。では、この難解なマシン語を少しでもわかりやすくするための「アセンブラ」について次に説明します。

マシン語とアセンブラ

マシン語は符号で、つまり数字です。2進数を16進数で表すことで長さは短くなるものの、プログラムを数字で書くなんてとてもじゃないけど普通の人間にはできません。

21 00 80

11 00 A0

01 00 10

ED B0

C9

なんて、ええい、なんでこんなもん覚えてんだよ。すみません。取り乱してしまいました。これマシン語でメモリの内容を転送するというものなんですが、こんな数字で書く人はあまりいません。通常は、

LD HL, 8000H

LD DE, 0A000H

LD BC, 1000H

LDIR

RET

のように書いて、翻訳プログラムに翻訳してもらいます。このように書く言語をアセンブリ言語、この翻訳を行うプログラムをアセンブラといいます。

アセンブラとマシン語は1対1に対応します。「JP」といえば16進数の「C3」だし、「LDIR」といえば「EDB0」です。だから、実際

にはマシン語の数字で書いているのと同じなんですが、足し算の命令を ADD、引き算を SUB などといった換えたほうが数を覚えるよりはるかに楽だろうと、そういうことからアセンブラが利用されているのです。

ビットとバイトと CPU

マシン語の話をするときに出てくる用語がいくつもあります。メモリとかアドレスとか、アセンブラとかもそうです。ビットというのは聞いたことはありませんか？ これ、2進数の1桁を指す単位なのです。2進数の8桁なら8ビット、16桁は16ビットですね。で、さらに8ビットは1バイトと呼ばれます。このバイトという言葉もおそらく聞いたことがあるでしょう。メモリが256Kバイトあるといえ、2進数の8桁を記憶できる領域が、256×1024個あるという意味になります。メモリの広さをいふときの「キロ」は1000ではなく、厳密には1024を表します。

CPU はマシン語を実行する、パソコンの要である部品のことです。マシン語はこの部品によって解釈され、この部品はほかの部品に指令を与える (といてはもちろん電気信号を1にしたり0にしたりすることで行います。この部品の内部で命令やデータを解釈する部分の回路が何ビット同時に検査することができるか、ということがこの部品の性能を大きく左右します。

これが「何ビット CPU」という表現の意味なのです。すべての PC-8800 シリーズに使われている Z80 という CPU は内部で8ビットを同時に検査しています。PC-9800 シリーズで使われている 8086 は16ビットを単位として「16ビット CPU」と呼ばれ、80386 は「32ビット CPU」というわけなのです。

というわけで、マシン語は何かということを書いてきましたが、わかっていただけたでしょうか。これだけでは原理はわかって、プログラムを書くにはいたらないでしょうから、次回には、具体的なプログラムを作る上での考え方を説明します。また、読者のみなさんの質問にもどんどんお答えします。

どしどし業書きでお寄せください。

What's 「MS-Engine」?

PC-Engineを MS-DOSの 動作環境に

先ごろ、NECニューウェイブシステム事業部から「MS-Engine」というユーティリティが発売されました。これがPC-88VAのユーザーのなかで注目を集めているのです。今まで開発ツールがなくVAユーザーはプログラムを作るに作れない環境にあったわけですが、この「MS-Engine」の登場によりMS-DOS上のプログラムをVAで動かすことができるようになったのです。これはちょっと耳よりな話ですよ、ね。

VAでMS-DOSの ソフトを動かす

PC-88VAのOSであるPC-Engineは、ご存じのとおりMS-DOSを強く意識しています。ディスクファイル形式が、高度な階層ディレクトリまで含めて共通なため、相互にファイル交換することが可能です、DOSのコマンドもほとんど同一です。

しかし、完全コンパチブルというわけではないので、MS-DOS上のプログラムを88VAで動かすことはできません。ところが、「NECニューウェイブシステム事業部」から発表された「MS-Engine」を使うと、88VA上でMS-DOSのソフトを動かすことができるということです。88VAユーザーにとって、これは見逃せないニュースでしょう。今回は、同時に発表されたビデオボード用ユーティリティ「G-turbo」と合わせレポートしましょう。

MS-DOSを見直す

16ビットパソコンといっても、各社からたくさんの機種が出ています。そして、NECのパソコンで動くソフトは、他社のパソコンでは動かすことができません（最近ではコンパチ機やエミュレーション機能を持ったものもあります）。なぜなら、パソコンごとにハードウェアが異なるからで

す。

たとえばあなたが知らない町に行ったら、セブンイレブンのある場所も、郵便局のある場所も分からず、右往左往するでしょう。

しかし、その町に精通したガイドさんがついてくれば、「本が買いたい」と告げれば、本屋に連れて行ってくれるか、希望の本を買ってきてくれるかをしてくれます。

パソコン内部もこれに似た事情で、キー入力する方法も画面に文字を表示する方法もパソコンごとに異なります。だから、別のパソコンに載せられてしまったプログラムは、お仕事をすることができません。

しかし、そのパソコンに精通したガイドに、「ディスクからファイルを読みたいんだけど」といえば、ガイドさんはその仕事を代行してくれます。このガイド役がMS-DOSと思ってください。

いまのはたとえ話ですが、MS-DOSはキーボードから入力するとか、画面に文字を表示するなどの出力に関する基本的な動作を行うサブルーチンを数多く持っています。このサブルーチンを呼び出して使うことをシステムコールといいますが、システムコールの方法がNECのパソコン上のMS-DOSと、他社のパソコン上のMS-DOSで統一されているのです。

システムコールが統一されているおかげで、違うパソコンでもMS-DOS上なら同じプログラムを動かすことができます（も

ちろん、それを意識して作ってあれば、です）。

しかしながら、グラフィック機能などは各パソコンで機能に大きな違いがあるので、MS-DOSで統一したシステムコールを作ることができません。だからMS-DOS上のプログラムだとしても、それがグラフィック画面やそのほか特定のパソコンの特殊な機能を利用してれば、ほかのパソコンのMS-DOSでは動かすことができません。日本語ワープロなどはその例です。し、PC-9801のMS-DOS上のN88-BASICなども、ほかのパソコンでは動きません。

だったらそういったプログラムはMS-DOS上で作る必要はないだろうといわれるかもしれませんが、しかし、MS-DOS上のワープロで作った文書ファイルは、MS-DOS上のほかのプログラムで読んだり利用したりできます。ファイルの互換性があるだけでも、計り知れないメリットがあるのです。PC-Engineが、MS-DOSとファイル形式を同一にしている理由もここにあります。

MS-Engine の意義

前置きが長くなりましたが、ここからが本題です。

PC-Engineは、ディスクに関係するシステムコールがMS-DOSと共通になってい

ます。

ところが、とくに画面への表示の方法がMS-DOSとは異なっているのです。だからMS-DOS上のプログラムをPC-Engine上で動かそうとしても、画面になにも表示されないなどの問題があります。

そこで登場するのが、今回紹介するMS-Engineで、ディスク関係以外の諸機能のシステムコールもMS-DOSとコンパチにしています。その結果、MS-DOS上の各種プログラムを88VA上で動かせるようになっていくのです。

前述の理由でPC-9801用の一太郎をMS-Engine上で動かすことはできません。どんなソフトを使えるかというと、MS-DOSの標準システムコールだけを利用して作られたプログラムです。形としては、キーボードからの入力、画面への文字の表示、ディスクアクセス、以上の3つを基本として作られた、比較的地味なプログラムです。

地味とはいっても、MS-DOSには各種の開発ツールや言語、ユーティリティが整っています。これまでのところ、88VA用には開発ツールなどないに等しかったわけですが、MS-Engineの登場により、一気に古参のパソコンに負けない開発環境がてき上がつてしまったことになるのです。

MS-Engineシステムを作ろう

MS-Engine Level 1を購入すると、ディスクが1枚ついてきます。このディスクの中には、表1のような9個のファイルが入っています。これらのファイルを、手持ちのPC-Engineのシステムディスクにコピーすると、それがMS-Engineのシステムに変身します。

具体的に、コピーの手順を説明しましょう。

すでにあるPC-Engineのシステムディスクを使うこともできますが、万一のことを考えて、新しいシステムディスクを作りそれをMS-Engine化します。

PC-Engineを起動して、AドライブにPC-Engineのシステムディスクを入れたま、Bドライブに新しいディスクをセットし、

FORMAT B: /S/V

と入力します。「FORMAT B:」によっ

て、Bドライブのディスクがフォーマットされますが、/Sオプションがついているため、AドライブのシステムがBドライブにコピーされて、新しいシステムディスクが作られます。また、ボリュームラベル（ディスクの名前）もつけるために、命令には/Vオプションもつけています。

ボリュームラベルは今回は、

MS-Engine

としました。

次に、MS-Engineのファイルをコピーします。

ドライブ1に、MS-Engineのディスクをセットして、

COPY A: B:

と命令し、表1のファイルをドライブBの新しいPC-Engineシステムディスクにコピーします。

これでOKです。こうして完成したMS-Engineシステムディスクは、オリジナルとして保存しておき、これをコピーして実用ディスクを作るとよいでしょう。

なお、普通のPC-Engineシステムディスクの初期フリー容量は、

1185792バイト

ですが、MS-Engineファイルをコピーすると、

1175552バイト

と、10240バイトほど減少します。この程度ならば実用上差し支えないでしょう（フリー容量は、システムのバージョンによって多少異なります）。

スタート方法

MS-EngineシステムディスクをドライブAにセットしてシステムリセットする

表1 MS-Engineの中のファイル

MSENGINE.COM	MS-Engine本体
BREAK.COM	MS-DOSの内部コマンド
VERIFY.COM	MS-DOSの内部コマンド
VER.COM	DOSバージョンの設定/表示
PRESET.COM	RS-232Cを初期化
AUTOEXEC.BAT	MS-Engine組み込み用バッチファイル
RS232C.COM	PC-9801用MS-DOSのSPEED.COMのアダプタ
COPYVA.COM	PC-9801用MS-DOSのCOPYA.COMのアダプタ
DUMPVA.COM	PC-9801用MS-DOSのDUMP.COMのアダプタ

と、MS-Engineがスタートします。

実際には、PC-Engineスタート後、表1中のAUTOEXEC.BATファイルによって、MS-Engine化が行われます。このファイルには、次のようなプログラムが書かれています。

```
CLS
MSENGINE
VER
PRESET
```

要するに、表1の中のMSENGINE.COMファイルを実行すれば、MS-Engineモードになるのです。そしてVER.COMファイルによってDOSバージョンを表示します。最後にPRESET.COMファイルによってRS-232Cポートをセットアップ状態にします。

MS-Engineの起動のためにAUTOEXEC.BATファイルが使用されるので、ユーザーが自動実行を利用したい場合は、上に示した4行に続けて必要な処理手順をプログラムすることになります。

付属コマンド

PC-Engineのコマンドは、MS-DOSのコマンドを基に作られていますが、そのすべてを持っているわけではありません。MS-Engineでは、次の7つのコマンドを外部コマンドとしてサポートしています。働きを説明しましょう。

・BREAK [ON/OFF]

各DOSコマンド実行中に、BREAKキー（CTRL-C）による処理の中断を許可するか、禁止するかのスイッチです。

・VERIFY [ON/OFF]

VERIFY ONにすると、以降ディスクの

書き込み時にベリファイを行うようになります。つまり、リードアフタライトモードとなるわけです。ただし、ディスクの書き込み時間が長くなりますし、ディスクの書き込みは十分信頼性が高いので、このコマンドを使う必要はほとんどないでしょう。

・VER

DOSのバージョンを表示、あるいは設定します。

単に、

VER

とすると、

MS-Engineバージョン3.10

と表示します。

以下の3つのコマンドは、PC-9800シリーズ用のMS-DOSの標準コマンドSPEED.COM、COPYA.COM、DUMP.COMの3つをMS-Engineで利用可能とするためのアダプタです。

・RS232C

RS-232Cインタフェースのボーレート、パリティなどの表示および設定をします。

・COPYA

補助入出力装置(ディスクドライブ、RS-232Cポート)との間でデータを送ったり、受け取ったりします。

・DUMPVA

ディスクファイルの内容を、16進数で表示します。TYPEコマンドの16進数版というわけです。

MS-DOSの開発ツール

MS-DOSには、開発ツールおよびユーティリティが統合していますが、それら有益なプログラム(形式的にはコマンド)はMS-Engineでも利用することができます。ただし、当然ながらMS-DOSのツールを所有していなければなりません。

MS-DOSの開発ツールを紹介しましょう。

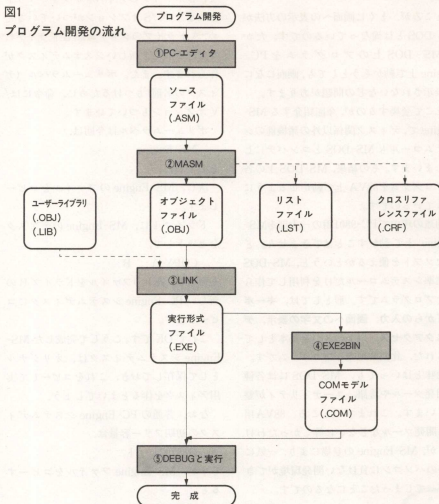
・EDLIN

ラインエディタです。MS-EngineはフルスクリーンタイプのPCエディタを利用できるので、EDLINを使う必要はないでしょう。

・MASM

8086のアセンブラですが、強力なマクロ命令機能を備えています。MASMがあれば、

図1
プログラム開発の流れ



ば、88VAも立派な開発システムとして活躍することができるのです。

MASMでアセンブラプログラムを開発する手順は、図1のようになります。

・LINK

MASMのオブジェクトファイルから、実行形式のファイルを生成します。同時にライブラリモジュールを組み込むことができます。

・EXE2BIN

COMモデルの実行ファイルを作る場合は、このEXE2BINを使用します。COMモデルとは、マシン語プログラム、データエリア、スタックエリアが64K(セグメント)内に納まるものですが、詳細は割愛します。

・CREF

アセンブルしたプログラムのクロスリファレンスを作成します。

・LIB

ライブラリファイルの作成、管理を行うためのユーティリティです。

・DEBUG

1ラインアセンブル機能と逆アセンブル機能を持った、比較的規模の大きなシン語モニタです。

・SYMDEB

DEBUGの機能を強化したものです。ブレークポイントなどの設定にソースリストのラベルを使用できるため、デバッグ効率が大幅に向上しています。

PC-Engineの内蔵シン語モニタは、このSYMDEBを意識して作られており、ほぼ同等のコマンドを使えます。ただし、シンボリックデバッグ機能は持ちません。

・FIND

指定ファイルの中から、指定文字列を含む行を探して表示します。

このコマンドには本来、指定文字列を含む行の総数を表示したり、行番号を表示するなどの機能があるのですが、それらはMS-Engineでは使用不能です。

以上のコマンドは、MS-Engineと共に

MS-DOSのマクロアセンブラを持っていない
れば利用できません。PC-9801用MS-DOS
のツールを購入することになりますが、価
格は18,000円。しかし、強力な開発シス
テムの価格としては、安いものです。

88VA とは別に PC-9801 と MS-DOS を
所有している方もおられるでしょうが、そ
の場合は大変お得です。

アプリケーションソフト

MS-DOS 用プログラムがすべて MS-
Engine 上で動けばこんなに便利なことは
ありませんが、初めに注意したとおり、使
えるのは標準 MS-DOS の機能だけを使用
して作られたプログラムに限られます。

また、MS-DOS のシステムコールの中
で、次のものはサポートしていないので、
これらを使用したプログラムも使用不可
能です。

- 0 DH (ディスクリセット)
- 0 FH~18H (FCB によるディスク
アクセス)
- 1 DH~24H (アクセス)
- 26H~29H (アクセス)
- 32H, 34H, 37H (非公開)
- 44H
- 50H~53H, 55H (非公開)
- 59H, 5CH 以上

MS-DOS 用の「Let's C」、*「Lattice C」*、
などの C コンパイラの動作は確認されてい
るそうです。ただし、「MS-C」は一部工夫
の必要があるとのことでした。

現在 NEC ニューウェーブシステム事業
部では、「Let's C」用のグラフィックライ
ブラを開発中だそうです。C 言語で 88VA
のグラフィック機能を使えるとは、おもし
ろいことになりそうですね。

いずれにせよ、MS-Engine は完全な
MS-DOS コンパチ環境ではなくエミュレ
ータですから、ソフト購入時には十分な動
作確認が必要です。たとえば今回手元にあ
った PC-9801 用 MULISP も、おそらく標準
MS-DOS 仕様と思ったのですが MS-
Engine では動かすことができませんでした。

なお、今回発表された MS-Engine は
Level 1 ということで、今後 Level 2、Level 3
と、より MS-DOS 互換性を強めたバージ

ョンに発展する可能性もありそうです。

「G-turbo」について

これは、MS-Engine と同様、NEC ニュ
ーウェーブシステム事業部から発表されま
した。PC-Engine または MS-Engine のコ
マンドモードで、スーパーインポーズやビ
デオデジタイズを実行できる、外部コマ
ンドパッケージです。それでは各コマンドを
紹介しましょう。

・SUPER ON/OFF

スーパーインポーズモードを設定、解
除します。

・ESC

テキスト画面を一時消去して、グラフ
ィック画面に表示しているビデオデジ
タイズ画像などを見やすくします。この状
態は、何かキーを押すと解除されます。

・GCLS

グラフィック画面を消去します。

・CLG [color code]

グラフィック画面を消去します。256
色のカラーコードで塗りつぶすこともで
きます。

・GRAPHIC ON/OFF

グラフィック画面の表示を許可、禁止
します。

・MOUSE ON/OFF

マウスカーソルの表示を許可、禁止し
ます。

・VIDEO ON/OFF

ビデオデジタイズを開始、停止します。

・SNAP

ビデオデジタイズモードを開始し、何
かキーが押されると、デジタイズをフ
リーします。

・VMODE [C]

グラフィック画面を、640×200 ドット
65,536 色モードに設定します。

C スイッチをつけると、モード設定後
グラフィック画面をクリアします。

・GMOODE [opt]

グラフィック画面を 640×400 ドットの
256 色モードに設定します。オプションと
して C をつけると、モード設定後グラ
フィック画面を消去します。

R, G, B をつけると、それぞれ暗い
赤、暗い緑、暗い青でグラフィック画面

を塗りつぶします。

・COLORS

256 色モードのカラー一覧表を表示し
ます。

・VSAVE filename

ビデオデジタイズ画面を、ディスクに
セーブします。

・VLOAD filename

VSAVE コマンドでセーブした画面を
ロードします。

・GSAVE filename

グラフィック画面をディスクにセーブ
します。

・GLOAD filename

GSAVE コマンドでセーブした画面を
ロードします。

各コマンドは、G-turbo のディスクから
毎回プログラムをロードして実行されるの
で、多少遅さを感じました。

なお、ビデオデジタイズおよびスーパ
ーインポーズ動作を伴うコマンドは、ビデ
オ出力モードにして使用します。

実際問題として、ビデオデジタイズやス
ーパーインポーズ機能を BASIC レベルで
制御することは容易なので、PC-Engine レ
ベルでこのような機能を実現する必要性は薄
いと思います。しかし、特別にプログラ
ムせずに手軽にビデオデジタイズができて、
それをディスクに保存できるというのはち
よっといいですね。

88VA 用のゲーム、ワープロなどが登場
し始めましたが、PC-Engine レベルでのア
プリケーションにも期待したいと思います。

また、MS-Engine の登場により、88VA
上での開発環境が整備されたことは非常に
喜ばしいことです。ますます 88VA が面白
くなってきた！

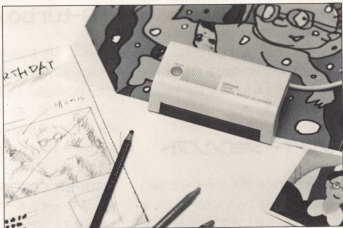
■「MS-Engine」 14,800 円
「G-turbo」 8,800 円
2 本のセット価格 18,800 円
■問い合わせ先 NEC ニューウェーブシ
ステム事業部
〒732 新居浜市久保田町 3-1-4 第 2
アイビル 1 F
TEL 0857(35)2280

88 GRADE UP TOOL

オムロンハンディイメージスキャナ 「HS10R」

後藤 清治

ハンディイメージスキャナが普及し始めました。これまで88対応のスキャナがほとんどでしたが、このたびPC-8800シリーズ対応のものがオムロンより発売されました。88ユーザーにとっては待ちに待った登場ではないでしょうか? 本誌ではこのマシン「HS10R」をとり上げ、パッチリ検証してみます。



普及する

イメージスキャナ

イメージスキャナ、いわゆる「画像取り込み処理装置」が市場に出回り始めたのは、4～5年くらい前ですが、今ではハンディタイプのものを含め、ワープロ、製図、ゲーム用と数多くのスキャナが販売されています。

では、なぜイメージスキャナのような装置がここまで普及したのでしょうか?

イメージスキャナの

歴史をたどる

スキャナが登場した頃は、「一太郎」に代表されるワープロソフトが一般に普及し始めた時期でもありましたが、そのなかでもPC-9800シリーズ用のものは早い時期から編集画面に絵を入れる機能をもっていました。

昔は、ワープロで作った文章に絵を入れる場合には、イラストや写真のコピーの切り張りをするしかなかったわけで、非常に体裁の悪いものでした。つまり、スキャナはもっと完成度の高いグラフィックスを、というワープロユーザーの要求によって生まれてきたものなのです。

それでも、昔のようにグラフィックスの解像度が160×100ドット程度だった頃はスキャナの必要性はほとんどなく、1ドットずつでも色を置いて絵を描くことは可能でした。

ところが、ハードの進歩により、グラフィックスの解像度は640×200、高解像度モードに至っては640×400ドットという高性能になったおかげで、昔のような悠長なこ

とはいつていられなくなったのです。

そこで、ユーザーは理想的な画像入力手段を求めて(つまり、いかに楽をするかを求めて)暗中模索することになったわけです。その結果、ユーザーのニーズは以下のように2つに分かれました。

1. 専用の画像入力デバイスパソコンに接続する方法
 2. ソフトウェアを使って絵を描く方法
- この2つの方法のうち、2の具体的な入力方法とは、

- ①ドットを1つずつ置いていく
 - ②紙に描いたものを座標データに変換する
 - ③特殊なカーソルを作り、ディスプレイに張りつけた絵をなぞる
- などがありますが、どれも時間と手間がかかり過ぎました。

そこで必然的に1の方法が選択されることになったわけです。

精度の高いイメージスキャナ

1の方法には、

- ①マウス
- ②デジタイザ、マジックスティック
- ③イメージスキャナ

の3つがあります。

①のマウスは、内部のボールを回転させて、その移動方向と移動距離を相対的に割り出して、パソコンにデータを送ります。

実際に絵の細かい修正にはよく使用されますが、マウス自体扱いが難しく、あまり適したものではありません。

②のデジタイザは、イメージスキャナと人眼を二分した入力装置です。マウスのよ

うな形をした入力装置とデジタイズボードで構成され、ボード上の1点を基準にした絶対座標をパソコン本体にデータとして送ります。デジタイザは読み取り範囲が小さく、値段が高いのが難点だといえます。

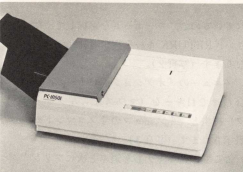
また、マウス、デジタイザの両者に見えることですが、原稿のなぞり描きやフリーハンド描きは可能ですが、写真の直接取り込みや精密な絵の複製は困難になります。

こういった問題を解決できるのがイメージスキャナなのです。

モノクロからカラーへ

多くの周辺機器がそうであるように、イメージスキャナも業務用として、印刷にコンピュータが利用されたことから発展しました。これがいわゆるCTS(注)化です。このCTSというのは、印刷する文章と絵の情報をデータとして記録しておき、ディスプレイ上でレイアウトして印刷する「DTP」(デスクトップ・パブリッシング)の大規模なものと考えていいでしょう。これは印刷の効率を飛躍的に高めることができるため、新聞などにもっとも利用されていました。

こういった業務用のスキャナは大型で、値段的にも個人では到底手の届くものではありませんでした。しかし、前述のように高解像度のパソコンの普及とともにスキャナの需要も高まり、廉価なものも登場するようになりました。NECの「PC-IN501」(写真1)もその中の1つでしたが、それでも298,000円という価格のため、パーソナル・ユースとはまだいい難いものでした。その後、スキャナ対応のソフトの数も増え、



100,000円前後のスキャナも出現し、その発展にはめざましいものがありました。

しかし、どれもモノクロ対応で、カラーや中間色のものには、ホワイトスクリーンというシートを使ってドットに分解して(2値化)読み込まなければなりません。

そして、とうとうユーザーの希望に応え、1986年、エプソンよりカラー対応のイメージスキャナ「GT-3000」が198,000円という手頃な価格で発売されました。このカラー対応のスキャナが現れて以来、ファックス機能を持ったものなど多機能で高性能なものが市場に登場するようになりました。

ハンディタイプの登場

1987年になると、ワープロ専用機も100,000円以下というパーソナル・ユースになり、同時にワープロの多機能化も各メーカーで争われ、ハンディタイプのイメージスキャナもその戦略上、重要な位置を占めるようになりました。そして相次いでハンディコピー機のノウハウを受け継いだハンディイメージスキャナが各メーカーから登場したわけです。テレシステムズの「HS-98 G」、エプソンの64階調の読み取りができる「GT-100」、同じくテレシステムズの間読み取りが可能な「HS-98mk II」、そして今回ご紹介するオムロンの「HS10R」です。

以上のようにイメージスキャナは大型機から小型機へと変貌を遂げていったわけですが、これまではPC-9800シリーズのものが大半で、PC-8800シリーズのユーザーは88対応のものを待望していました。

では、待望の88対応のこのスキャナ、さっそく使ってみましょう。

機能概要とマシンの外観

「HS10R」の外観は、写真を見ていただければわかりますが、軽量で持ち運びやす

図1 「HS10R」の外観

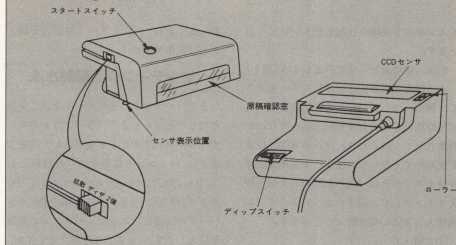


表1 「HS10R」の仕様

読み取り方式	インタフェース
密着イメージセンサ	RS-232Cインタフェース
スキャナ手動走査方式	(1)9600/19200 BPS
階調	(2)8ビットパリティなしストップビット1
最大延誤64階調以上	制御コマンド手順
(1)2階調単純出力	PC-I N511/503(501/502)に準拠
(2)16階調ディザ法	読み取り線密度(拡大・縮小)
(線点, BAYER, ウズマキ)	90, 120, 160, 180, 200, 240, 400DPI
(3)64階調以上 誤差拡散法(重みづけ)	(45~200%)
(タイプI文字, 線画, 写真の混在に優用。タイプII線点印刷用)	外形寸法
読み取り幅	130(W)×70(D)×41(H)mm
104mm	接続ケーブル長
読み取り速度	1.5m(コネクタ, ケーブルは標準)
約3msec/Line(A6サイズ, 約4秒)	重量
読み取り分解能	300g(スキャナ部のみ)
(1)主走査(幅) 8ドット/mm	電源/消費電力
(2)副走査(縦) 8ドット/mm	9V/専用ACアダプタ使用)
(3)有画面素数 832ドット/line	15W以下
ドロップアウトカラー	環境条件
実録	(1) 温度 動作時0~40℃保管時-20~60℃
入力バット	(2) 湿度 動作時35~80%保管時20~90%
120Kバイト	(3) 直射日光, 直射球下での使用を避けること

表2 A ディップスイッチの機能

No.	ON/OFF	機 能
1	ON	19200BPS(19200BPSをサポートしているパソコンのみ)
	OFF	9600BPS(初期値)
2	ON	SW2 SW3 SW4 線密度(DPI)
	OFF	OFF OFF OFF 200(初期値)
3	OFF	OFF ON ON 90
4	OFF	ON ON OFF 120
	ON	ON ON ON 160
	ON	ON OFF OFF 180
	ON	ON OFF ON 200
	ON	ON ON OFF 240
	ON	ON ON ON 400
5	ON	黒地に白細線のOR処理になる
	OFF	白地に黒細線のOR処理になる (初期値)
	ON	誤差拡散(II) ディザ(Bayer)
		印刷物などで網点処理が施してあるものに對してモアレ除去を行う トの強い表現
6	OFF	誤差拡散(I)(初期値) ディザ(線点)(初期値)
		濃淡の重みづけのある誤差拡散法 新聞写真などで使われている方式
		写真, 線画を最も忠実に表現 ディザの中では最も自然な表現

図2 ディップスイッチの各モード

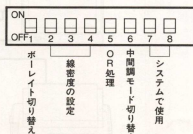


表2 B

読み取り線密度(DPI)	倍率
90	45%
120	60%
160	80%
180	90%
200	100%(標準)
240	120%
400	200%

くスッキリと洗練されたデザインになっています。

では、機能面から見てみましょう(図1)。本体側面の原稿確認窓は見やすい大きさで、側面についているセンサ位置表示の突起が原稿のどのあたりに読み込み中なのかが判断されてくるため、読み込み開始位置が決まやすくなっています。本体側面の切り替えスイッチは「読み取りモード切り替えスイッチ」で、以下のように3タイプの画像入力方式の切り替えができます。

①2値入力

これは、原稿を白と黒の2値で判断し読み取るモードで、文字、ロゴ、地図、手描きの絵など、濃淡のない図形に適しています

②ディザ

写真などの中間調を濃淡のブロックで表現します

③誤差拡散

顔の輪郭や起伏、陰影のある写真やイラストなどをよりリアルに表現します

また、装置の反対側についているのは、原稿の読み取り濃度を調整するボリュームです。これは、すべての読み取りモードで有効です。裏面についている左右のローラーはスキャナの移動を検出します。

本体後部には、ディップスイッチがあり、いくつかのモードに切り替えることができます(図2)。まずディップスイッチ1は、ポートの切り替えスイッチで、ONで19200ボー(FH, MHなどの19200ボーをサポートしているパソコンのみ)の通信モードになり、OFFのときは9600ボーになります。スイッチ2, 3, 4はON/OFFの組み合わせ方で、8とおりの線密度の違いはそのまま読み取り倍率の違いとなって現れます。スイッチ5は、ONで黒地に白細線のOR処理、OFFで白地に黒細線のOR処理になり、絵の性質によって使い分けます。

スイッチ6は、ONで誤差拡散II(網点処理の施してある絵などを取り込む場合に適します)、OFFで誤差拡散I(濃淡の重ねつけができ、写真、線画の取り込みに適します)になります。ディップスイッチ7, 8は使用時は必ずOFFにしておきましょう。

このスキャナはNECのプリンタPC-IN501/502/503/511シリーズのRS-232Cインタフェースデータ転送手順と互換性が

ありますから、INシリーズ対応の市販ソフトは使用可能です。

「セッティングと起動方法」

まず、パソコンの通信モードを表2を参考にしてセットします(各パソコンのユーザーマニュアルなどを参考にして設定します)。パソコンの電源を切った状態にして、RS-232Cにスキャナからコネクタでつなぎます(図3)。コネクタに付属のDCコンバータを接続したあとに、パソコンの電源を投入します。いうまでもないことですが、スキャナのディップスイッチ1もパソコンの通信モードに合わせてセットしてください。使用するソフトは今回はタイトのZ's STAFF Kid 88です。

このソフトはお絵描き用で作画はマウスやイメージスキャナを用いて行えます。

また、作画したものに着色や縮小、拡大がかけられ、オリジナルなコンピュータグラフィックスが楽しめるようになっていきます。

「Z's STAFF Kid 88」 で使ってみよう

このハードを使用して一番感じたことは名前のおと「ハンディ」そのものだという事です。取り込み中に本体を手前に引くわけですが、移動用ローラーがなめらかに動き、原稿確認窓が大変見やすいため使い勝手がよく、機動力の高いものになっています。

それではソフトを実際に使用した場合のイメージスキャナの画像取り込みの様子をお見せしましょう。

スキャナをセットして、ソフトのシステムを起動します。デバイス選択でキーボードを選択して(JOYマウスを持っている方はそれで大丈夫ですが、シリアルマウスを持っている方はRC-232Cをスキャナに取ら

れているので、キーボードを選択してください) RETURN キーを押すと初期画面になります(写真2)。

メニューの「オプション」にカーソルを合わせてF1キーを(マウスなら左クリック) 押すと、オプションのサブコマンドが表示されます(写真3)。

ここで「スキャナ」を選ぶと次にプリンタの機種名を聞いてきます。機種は3種類

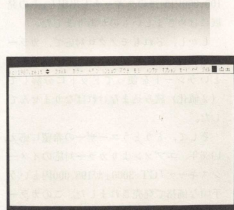


写真2 「Z's STAFF Kid 88」の起動画面

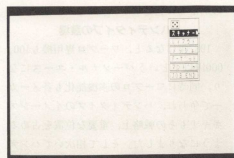
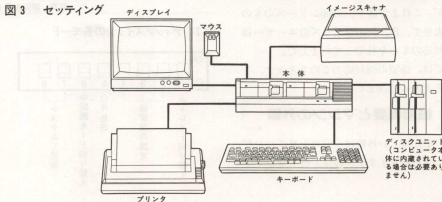


写真3 オプションのサブコマンドを表示した画面

表3

調歩同期	9600BPS/19200BPS
データ形式	データ長 8ビット
	ストップビット 1ビット
	パリティ なし

図3 セッティング



ディスクユニット(コンピュータ本体に内蔵されている場合は必要ありません)



図4 ゴリラのイラストの原画



写真4 順次拡散方式で取り込んだゴリラのイラスト



写真5 ディザ方式で取り込んだゴリラのイラスト



写真6 2値方式で取り込んだゴリラのイラスト

は画面の左半分が表示用に使われます。

目的にあった 使い方をしよう

さてそれでは、

①誤差拡散方式

②ディザ方式

③2値方式

のそれぞれのモードで同一の絵を取り込んで、その違いを見ることにします。サンプル画としてはオムロンのスキヤナ本体についてくるゴリラの絵を使用します(図4)。

まず、写真4は①の方式でスキャンしたものです。このサンプル画は陰影の濃い精密なリアリティのあるイラストですので、この方式は最速といえるでしょう。実際、図4の元原稿と見比べていただければよくわかると思います。

次に写真5の②の方式ですが、サンプル画に中間色のトーンがあまり使われていないために①ほどの表現力が見られませんでした。

写真6の③の方式は最も表現力が弱く、中間色のまったくないベタリした感じになっています。

ほかには私が趣味で描いているイラストを③の2値方式で取り込んでみましたが、このタイプのものもやはり①がよいようにでした。

このスキヤナを何度となく使ってみてその使い心地のよさに内心驚いていたのですが、こんな装置でも多少なりともいい絵を作るための条件があるようです。

まず場所。これはものをちらかしていない平坦な机。いくらこのハンディスキヤナがコンパクトでも、山のようにものを積み上げている場所ではスペースの確保もできません。

次に、取り込みたい原稿の質。原稿が薄くてばやけているといくらスキヤナの性能がよくてもいいものはできません。もし、仮に元の原稿が薄い場合は、複写機で濃めにコピーをとって目的のものをはっきりさせることが必要でしょう。

「HS10R」 をチェックする

イメージスキヤナの歴史をたどり「HS10R」についていろいろ試してみました。

本体の手軽さはやはり特筆すべきものがあります。本体の重さが300グラムですか

ら普通のマウスの約2倍程度の重さです。画像の取り込みには結構時間がかかるものですから、この程度の軽さだと長時間使用してもあまり疲れません。

それからコンパクトな設計による機動性のよさ。細かい図面でも小回りが効くので絵が途中で切れたり、思うように絵が入らないということがありません。

そして機能面とは別に、対応するソフトウェアの数が多いこと。この柔軟性は大きな特典ではないでしょうか。

しかし、よいことばかりでもありません。次にあげるようないくつかの問題も残っています。

まず読み取り速度ですが、3m sec/lineで、これは原稿がA6サイズで約4秒の時間です。

この速度でいざ動かすと、意外と同じ速さで移動させていくものなのです。移動速度が速すぎると、イメージセンサの読み取りが追いつけなくなるため、追いつけない分のデータはすべて取りこぼしてしまい、ところどころ絵の抜けたベシヤンコの画像になってしまいます。

マシンを移動させる速度の目安は、葉書きの上から下までを「1」、「2」、「3」、「4」とゆっくり数えながら動かすとちょうどよいでしょう。

次にローラーの移動方向ですが、力の加減を間違えると、途中で斜めに読み取ってしまうこともあります。これはマシンの安定度の問題ですが、不安定なためにマシンを握って手前に引くとき、少し緊張してしまいます。もう少し安定感がほしいというのが正直なところですね。

総合的に見てこのイメージスキヤナの性能、使用感にはすばらしいと思います。価格的にもリーズナブルで、これならユーザーの要求を十分満たしてくれるものだと思います。ハンディスキヤナを利用してオリジナルな写真入りや年賀状やイラスト集など、自分のライブラリを作れば大変おもしろいと思います。

あなたも手持ちの88をゲームやプログラムのためだけでなく、アートギャラリーとして開放してみたいかがてしょう。

「HSIOR」 45,800円

注 CTS(Computerized Typesetting System)
電子計算機を用いる写真組版方式

に分かれますが、今回は「IN-501/502/511」を選択します。そこからさらにサブ画面が開かれて、取り込む図形の色(8色から1色選択)と背景の色を選択します。画面範囲「640×200」を選んだあと、「実行」を選択するとスキヤナが準備されます。

準備段階のスキヤナは中でライトが点滅します。そして、本体の「READ」スイッチを押すとライトが点灯したままになって読み取りを開始します。一定の速度でスキヤナを手前に引き、目的の画面上をなぞります。

取り込んだ画像は順次ディスプレイに表示されます。読み込み画面の大きさは葉書きを縦に置いたくらいです。ディスプレイで

P.1で遊ぼう①

おもしろグッズにTRY!

IMAGE TURBO 88



昨年10月16日に発売されて以来、注目を集めているデービーソフトの「P.1」。本誌では前号でそのデスクトップ・パブリッシング感覚をすでにお伝えしましたが、今回はP.1対応のハンディイメージスキャナ、「IMAGE TURBO 88」を使っておもしろグッズを作って楽しんでみようと思います。顔写真入りの名刺、カレンダー、リフィル住所録と、ワクワクするようなものばかりです。

猪久間 達也

P.1は、昨年発表されたPC-8801およびPC-8800シリーズ向けの実用的なソフトウェアの中で、最も話題を集めたものです。

パソコンで出版を行うデスクトップ・パブリッシングの波をとらえながら、日本におけるレーザープリンタの普及率やフォントの問題を考慮して現実的なレベルでまとめあげ、しかも8ビット機であるPC-8800全機種（およびPC-88VA）で動くようにしてしまったことは、新機種優先のソフトウェア業界にあって一服の清涼剤のように感じられました。実際の製品の滑り出しも好調と聞きます。

今回は、このP.1とハンディなイメージスキャナである「IMAGE TURBO 88」を使って、パーソナルパブリッシングの一端をのぞいてみることにしましょう。

まずP.1について

P.1については、すでに広告や「Oh! PC」の記事でも紹介されていますからご存じの方も多いと思いますが、念のため簡単に説明しておくことにします。

P.1はゲームソフトからワープロまで広範囲に手がけているデービーソフトの製品で、一言でいえば、ワープロの次にくるも

のといったイメージのコンセプトを持っているソフトです。

具体的には、8ビット機最強、16ビット機でもどうかと思われるほどの表現力を誇るワープロをベースに、マウス対応のモノクログラフィックエディタを内蔵しています。さらに独立したカラーグラフィックエディタも付属していて、文書作成に関する基本的な部品が1つのパッケージに収められているといった印象です（価格は40,000円）。

これを原稿作成、編集、印刷というパブリッシングの基本に照らして考えてみましょう。

まず文章入力力はAXS（Advanced Transration System）と呼ばれる逐次自動変換システムを搭載して漢字入力時の手間を省いています（通常の重文節変換も可能です）。もちろん辞書には学習機能があり、使い込むほど変換効率がアップします。約20回程度までさかのぼって確定漢字の再変換ができるのも、画期的（8ビット用では最初の機能）です。

グラフィックスもP.1の特徴的な部分ですが、画面切り替えをすることなく、文章と同一画面で入力ができるようになっていきます（内蔵のモノクロエディタ使用時）。ま

た、一度描いた絵はグラフィック外字と呼ばれる特殊な文字に変換されるため、複写や移動や削除など文字とまったく同じように扱うことが可能です（グラフィック外字の部分は、何度でも絵にもどして編集することができます。さらに不用意に編集されないように図形や枠の保護機能ももっています）。

制限としては、1文書当たりのグラフィックスの分量が表示画面の約3.3倍までで、絵からグラフィック外字（またはその反対）への変換に多少時間がかかることがあげられますが、パーソナルユースとしては十分ともいえるでしょう。ただし、印刷時の字間、行間の設定によっては図形が歪むことになるので、注意が必要です。

編集に関しては、枠線を利用した疑似的な「段組編集」や「ブロック編集」ができるほか、「右詰め」、「左詰め」、「均等割り付け機能」をもち、バラエティに富んだ紙面構成が楽しめます。表現力を左右する文字の装飾機能についても、「網かけ」、「反転」、「下線」、「斜体」、「緑どり」、「強調」、「カラー」、「開目」、「回転」、「鏡文字」と豊富にそろっており、ほかに何が必要かと思うほどです。

さて、いくら画面がすばらしくとも、印

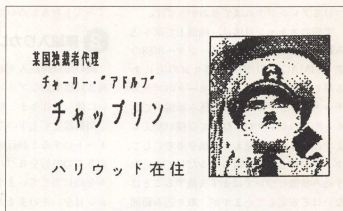
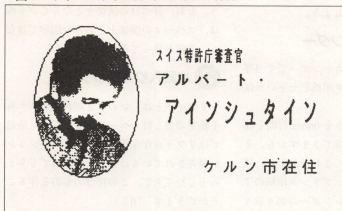


写真2 IMAGE TURBO 88



刷との落差があつては何にもありません。P.1は話題のレーザープリンタも使えるようになっているほか、市場にあるPC-8801および8800シリーズ向けの漢字プリンタのほとんどに対応しており、印刷用の漢字フォントを2種類（明朝体とゴシック体）用意して、装飾文字も美しく出力されるようになっています。

もちろんドットインパクトプリンタの出力を、そのまま印刷用の版下にするには無理がありますが、社内文書や学級新聞などには十分なクオリティを備えているといえるでしょう。

P.1のパートナー IMAGE TURBO 88

今回、P.1のパートナーとして使った「IMAGE TURBO 88」（以下イメージターボ88）は、手のひらサイズのイメージスキャナです。少なくともパーソナルパブリッシングというからには、作成する文書にはグラフィ

ックスをふんだんに入れたいものです。

しかし、絵はどうしても苦手という人や、写真を取り込んで使いたいという場合には、単なるグラフィックツールだけではどうにもなりません。こんなときに電子的なコピーマシンともいえるスキャナがあれば、気に入ったイラストや写真原稿をそのまま取り込んで使うことができるのですが、つい最近までスキャナは価格的にもコピーマシンと同じように高く、なかなか個人で手の出るものではありませんでした。

ところがポータブルタイプのワープロの普及と共に、取り込める幅は狭いものの、手軽に使えるハンディスキャナが登場し、パブリッシングの世界がぐっと身近になったのです。「イメージターボ88」は有効取り込み幅が64mmで、単純2値とディザ法による3種の階調表現をサポートしているハンディスキャナとしては標準的な製品で、価格も34,800円とリーズナブルなものになっています。

オリジナルグッズに 挑戦!

① 顔写真入りの名刺

P.1はもともとワープロにできるようなことであれば、何でもこなすことができます。いわゆる「大物」のニューズレターなどに関しては広告でもおなじみなので、ここでは遊び心を生かした「小物」に的を絞り、身のまわりにあるパブリッシングの実例としてのP.1の試作例をお目にかけましょう。

まずはじめは、顔写真入りの名刺です（図1）。ビジネスでも、熱心な営業マンなどは、自分を印象づけるために顔写真を刷り込んだ名刺を持ち歩いていますが、P.1とイメージターボ88の組み合わせせならば、多少鮮明さはなくなるものの、簡単にオリジナルの名刺ができてきます。最近の中高生の名刺ブームでも、友だちに差をつけるよいチャンスでしょう。

①手順としては、名刺の大きさが90mm×55mmですから、P.1の用紙をフリーサイズカタックシールに設定して最低幅の100mmを入力し、一度文字を並べて試し打ちをします（字間8ドット、行間16ドットのときに、画面上のイメージとプリントアウトの比率がほぼ一致します。実際には端数が出て、ピッタリこの数字にはなりません。が、入力後±1ドットくらいのおさめればOKです）

②次に、プリントアウトを参考に文字の位置と図形領域の大きさを決めます。そして図形編集モードのスキャナ入力を使って用意しておいた顔写真を取り込み、枠などを

つけてアレンジすればできあがりです。

実際のスキャン自体は、画面上で取り込み範囲を指定した後、イメージターボ88の窓("IMAGE TURBO"のロゴの上にある黒く細長い四角が実はスモークのプラスチックでできており、取り込み準備ができると内部のランプが点灯して原稿が見えるようになる)から見える原稿を参考に、ボタンを押しながら滑らせるだけです。取り込み状態についてはまず失敗することはないほど安定していますが、取り込み範囲のずれは自分で取り直しすることで調整してください。

もう1つ注意しなければならないのは、スキャナによる拡大率です。イメージターボ88とP.1の組み合わせの場合、長さで2倍強、面積的には4倍以上に拡大されますから、取り込み写真原稿は結構小さいもので大丈夫ということになります。

実際の自分の名刺ばかりでなく、図のような冗談名刺をパーティなどのために作っ

てみておもしろいでしょう。

2 挿絵入りカレンダー

続いては挿絵入りのカレンダーです(写真3)。標準的なプリンタ用紙を上下いっぱいを使って作りました。

用紙設定で上下の空きを0mm、行間を16ドットにすると48行確保できますから、そのうちの31行を日づけ、2行を見出し、残りを絵に当てています。プリンタ用紙のミシン目を、そのままカレンダーの切り取り線にできるところがミソです(上下ギリギリなので、とじ代を作るためには1枚おきに印刷しなければなりません)。

挿絵は北斎や広重などの版画から取ってしました。パーソナルパブリッシングと日本画も意外と相性がいいな、といったところでした。

画面上では、土曜日を青、祝祭日を赤で表していますから、カラープリンタで打ち出すと一段とカレンダーらしくなるでしょ

う。なお、日づけの区切りとなっている線は、スペースの関係上、縦線や図形ではなく、下線を使って書いてみました。

3 地図入り住所リフィル

リフィルとは、いわずと知れたシステム手帳用の差し替えページのことです。最近ではリフィル作成専用のアプリケーションが発売されているほどですが、P.1でもちょっとした工夫で、この程度のものを作ることができます(図2)。

リフィルの大きさは95mm×170mmですから、用紙をフリースizeとして横を100mm、縦を170mmにセットします(さきほど書きましたがP.1の最低用紙幅は100mmなので)。次に枠線を使って大まかな区分け(パンチ穴をあけるためのスペースを忘れずに)をして、名称や住所などの間の線は例によって下線を使ってスペースの節約をしています。

リフィルでは、文字の代わりに記号を使って項目を作ったりすることがありますが、ここでも電話マークで使っているように、P.1には豊富な外字パターンがあるため、その点でも活躍しそうです。

地図の部分は、2値処理で取り込んだあとでグラフィックエディタで画像の整理をし、駅名と会社名、マークなどは書き加えました。「九段南」や「北の丸」などの字はほぼそのままです。

なお、ここではフォントディスクを使い分けてプリントアウトをほかにも作ってみました。これだけでも結構雰囲気が変わるものです。リフィルなどはパーソナルな印刷物の代表のようなものですから、このようなバリエーションが楽しめるのもうれしい機能だといえるでしょう。

P-BASE 1とは何か?

最後に何かP.1のおもしろい使い方はないかと考えて、簡単なプレゼンテーションや研究発表に使えるP-BASE 1を考えてみました。この名前はベストセラーのデータベースソフトdBASEシリーズからいただいたものですが、P.1を簡易図形データベースとして使ってしまうというものです。

その方法ですが、写真4を見ればおわかりのように、各ページの頭に1画面に入り

図2 地図入り住所リフィル

リフィル住所録	
名称	日本ソフトバンク 出版部
住所	〒102 千代田区 九段南2-3-26 井関ビル
☎	03(237)7895
備考	裏手に本社あり

MAP	
交通 地下鉄新宿線・東西線 九段下駅より徒歩5分	

写真3 挿絵入りカレンダー



きる範囲で説明図と文字情報をまとめ、最上段にそのページのキーワードを書きおきます。これでP.1の「前頁」「次頁」機能を使い、次々に必要な画面を呼び出して説明に使えるようになります。また、キーワードを入れたことにより、P.1の「検索」機能を使って目的の情報を瞬時に取り出すこともできます（最上段に入れたのはP.1は見つけた単語が画面の一番上になるように表示するので、そのときに情報がきちんと見えるようにするためです。また、「検索」機能を使用中は、検索する単語の入力のために1行取られてしまうので、それを考慮して画面上の情報は15行で抑えておく必要があります）。

欠点としては、大きな絵はあまり多く入れられないこと（とくにRAM容量の小さなマシンでは、印刷はできても画面表示ができない場合がある）と、検索の途中で余計な単語を拾ってしまうことがあげられますが、画面構成とキーワードの選び方に気をつければ結構おもしろく使えるはずです。

読者のみなさんも、これ以外にいろいろ工夫して、自分だけのP.1の使い方をどんどん見つけていってください。

今後のP.1

さて、P.1は先頃「P.1 TERM」(後述)が発売されてシリーズ化されました。今後の予定はまだ明かではありませんが、グラフィックソフト、スプレッドシートなども発売されるのではないかと考えられます。その場合には、当然P.1と同じ日本語環境が搭載され、P.1ユーザーには違和感なく使えるようなものになるはずです。また、P.1から見れば使用可能なデータの幅が広がるわけで、数値処理が可能になることでビジネスユースへの適合性も高まることでしょう。

8ビット機の新たなソフトウェア環境を作りあげるものとして、今後のP.1シリーズに期待したいものです。

P.1 TERM

P.1 TERMはP.1の姉妹品で、P.1が新しいワープロの形を目指しているように、新しい通信ソフトを目指した製品です。

今までの通信ソフトはどちらかといえば

マニュアルで、初心者にはとっつきにくかったのですが、P.1 TERMはファンクションキーとカーソルキーによるメニュー選択方式を大幅に取り入れて、操作の簡素化を図っています。

主な特徴としては、P.1と同じ漢字変換方式を取っているで、P.1ユーザーにとっては新しい変換方法を感じることもない。使える点をはじめ、内蔵エディタによるテキスト編集の効率化、主要なBBS局をファイル化してワンタッチで呼び出せるようにしていること、受信時の漢字コード自動判別、通信中でも可能なスクロールバックなど、高機能と使いやすさを両立させようとした機能選択が光っています。

このほかにも、ファンクションキーに対する短文登録や、自動運転を可能にするオートパイロット機能など、ベテランユーザーが重宝しそうな機能もあり、クラス最強の通信ソフトといっても過言ではないでしょう。

もちろん、PC-8801(および88)シリーズ全機種対応というのもP.1シリーズならではのうれしい配慮で、P.1とコンビで使えば、文章ばかりでなく絵のデータも転送できるようになっています。

難点は、P.1のシリーズとはいえないが、操作の不統一なところがある(P.1はシステムをドライブ2に入れるが、こちらはドライブ1に入れる。明白な終了メニューがない、など)ことです。しかし、画面上のメインメニューにはマニュアルのページ数が明記してあり、そのマニュアルのてきがよ

いこともあって、操作中はあまり問題にならないと思われます。

P.1で楽しむおもしろグッズ

データベースソフトではP.1のユーザーのために会報誌を発行しています。

ユーザーの情報交換の場としてユニークなアイデアによる作品が発表されています。ここに挙げるのはその中のほんの一部ですが、本文でもご紹介したリフィルはなかでもやはり、今一番人気があるようです。

下の写真5の⑧は1988年のリフィル用のカレンダー。今年で印刷もできて、横のスペースに今年の目標などを書き込むこともできます。

⑨は同じくリフィルの月間予定表です。これなら、オリジナルのシステム手帳も思いのままです。

ほかにも、ビデオの整理には打ってつけのVIDEO DATA表があり、これはタイトル、キャスト、スタッフ、タイムなどのデータが書き込めます。

●問い合わせ先 データベースソフト
〒060 札幌市中央区北1条西7 住友海上札幌ビル TEL 011(251)7462

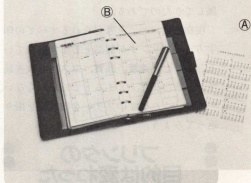
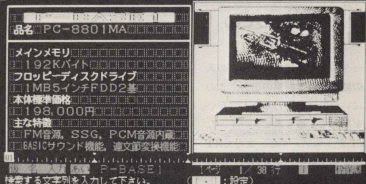


写真5 P.1で作ったリフィル

写真4

■キー:8801, MA, 192K, 1MB, FM, SSG, PCM, 通文変換機



検索する文字列を入力して下さい。

(F1:設定)

プリントアウトユース

から

DTPツールへ 「プリンタ」

1976年の「TK-80」の登場以来、12年あまりで32ビットパーソナルコンピュータが現れ、この急速な変化にはただ目を見張るばかりである。このシリーズでは、毎回テーマごとにその著しい変化を機能、時代を追って解明していく。今回はただの記録用紙から、デスクトップ・パブリッシングのツールとして飛躍した「プリンタ」を取り上げる。

木内 等

めまぐるしい変化をとげたプリンタ

パソコンの周辺機器といえば、まず思い出すのがプリンタ。

プリンタの歴史を見ると、「高速化」、「高品質化」、そして「コストダウン」に象徴される。つまりこれらを満足させるためいろいろな印字方式が考案され、それぞれに発展してきたのである。

現在では、およそ考えられるすべての印字方式のプリンタが製品として登場している。しかし、速度、品質、コストの改善競争は、まだまだとどまることを知らない。

では、これまでのプリンタの歴史を振り返ってみよう。

プリンタの 目的は変わった

プリンタは、基本的には文字を印刷する機械である。しかし、その用途や目的が時代とともに大きく変化してきたことは意外に認識されていない。その変化を時代を追って確認してみよう。

1. ディスプレイとしてのプリンタ

コンピュータによる業務は、データの入力、データの処理、処理結果の出力、この3点。これはいつの世も変わることはない。

処理結果を出力する最もポピュラーな機器は、CRT ディスプレイだ。またCRTの表示によってキーボードからの入力を確認できるため、入力装置として考えることもできる。

しかし初期のコンピュータは（大型コンピュータであっても）、CRTがつかうなかった。そのためのハードウェア技術がまだ不十分だったからである。

だからコンピュータの入力、出力機器としては、紙に文字を印刷するもの、つまりプリンタしかなかった。プリンタは、初めはCRT同様の「ディスプレイ装置」だったというわけだ。

現在においても、オフィスで見かけるテレックス端末はCRTなど持たず、キーボードからの入力は同時に印字される文字によって確認している（最近ではCRTと編集機能を備えたテレックスも登場しているが、いずれにせよファクシミリ通信に取って代わられる運命にあらう）。

2. データ記録のためのプリンタ

マニアがボードマイコンを自作し始めた時代、ディスプレイ装置として「7セグメントLED」が最もポピュラーだった。このちがはな表示装置によって、16進キーボードからの入力を確認し、メモリの内容を覗き、処理結果を表示していた。

しかしながら、何百ステップものプログラムを1バイトずつ確認していたのでは面倒で仕方がない。紙に印字すればプログラムを一度に見ることができる。実行結果も残したい。だからプリンタが欲しくなる。

その後パソコンの時代が来た。パソコンはボードマイコンと違って初めからCRTがつかう。画面も80文字×25行で、一度に2000文字も表示できる。このため、プリンタはなくてもとりあえずプログラムを作

るのに困らなかった。

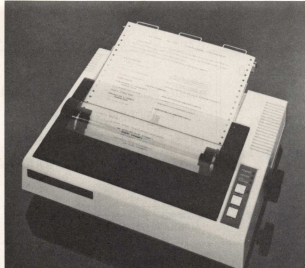
しかし、画面上の文字はCLRキーを押したり電源を切つてしまえば、一瞬に消えてしまう。一方、プリンタで印刷したものは半永久的にいつでもすぐに読むことができる。あたりまえだが、「目で見る形で記録を取る」ということが、プリンタの基本目的なのである。

とくにビジネスパソコンにおいては、結果を印刷することが重要。このため、プリンタを本体に内蔵したビジネスパソコンが登場したくらいである。

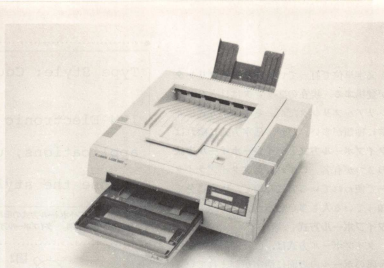
3. 高密度ディスプレイとしてのプリンタ

パソコンも初めは文字だけしか表示できなかったが、ハードの進化に伴ってグラフィックスが扱えるようになり、必然的にグラフィックイメージを印刷できるプリンタが登場した。

パソコンのグラフィックスも、初めは解像度が粗かった。たとえば初代のPC-8001のグラフィックスは160×100ドット。これは印影処理をした3Dグラフィックスを表示させるには不十分だ。そこで内部的に計算した高密度な仮想図形を、CRTではなくプリンタに打ち出すという手法が用いられた。それは、この頃のグラフィックプリンタ（写真1のエプソン「MP-80」など）が、1行に数百ドットのビットイメージ機能をすでに備えていたためだ。一度はCRTにディスプレイとしての立場を奪われたプリンタだったが、再び「高密度ディスプレイ装置」として見直されたのである。



●写真1 エプソンの初期のグラフィックプリンタ「MP-80」



●写真2 キヤノンのLBPシリーズの最新鋭、レーザーショット「LBP-B406」

4. 英文書印刷機としてのプリンタ

パソコンの用途の1つに、ワードプロセッシングがある。以前のパソコンは漢字が扱えず、ワープロといえば英文ワープロを指す時代があった。

英文ワープロで作成した文書は最終的にプリンタで印刷するわけだが、問題は文字の品質である。

何しろ当時のドットプリンタは、1文字を縦7×横5ドットなどで構成しており、プログラミストならともかく、文書として通用する文字品質などなかった。公文書作成用プリンタとしては、タイプライターのような活字方式がよく使われていた。

5. 日本語印刷機としてのプリンタ

パソコンの性能アップもとどまるところを知らない。とうとう漢字を使える時代がやって来た。当然、プリンタも漢字を印字できるようにになった。

ワープロといえば日本語ワープロのこと。それを使うためにパソコンを買い、和文を印刷するためにプリンタを買う時代になったのである。

6. グラフィック画面コピーのためのプリンタ

パソコンはすでに高解像度のカラー画面を使えるようになっていた。グラフィックツールもたくさん発表され、絵心のある人はCRTをキャンバスにして絵の作成にいそしんだ。画面に絵が描ければ、紙にハードコピーしてみたくするのが人情。グラフィックプリンタに、階調つきのモノクロで

印刷することもできたが、やはり色つきでそのまま印刷したいという要求は強かった。それに応えてカラープリンタが登場した。

7. 本物の印刷機としてのプリンタ

これからは、ワープロではなくデスクトップ・パブリッシングの時代だ。そしてデスクトップ・パブリッシングは、本当の印刷機としてのプリンタを必要とし始めた。そこで登場したのがレーザービームプリンタ(写真2)。

現在のプリンタは、文書もカラー図版も一度に打ち出せるし印字品質も速度もよくなっているが、まだまだ本物の印刷機の品質と速度には到達していない。フルカラーのレーザービームプリンタで安価なものが、一応理想形といえるだろう。

●メカニズムの進化●

プリンタの印字方式の分類を図1に示す。印字方式を、だいたい登場した順に解説しよう。

活字方式

ひとくちに活字方式といっても、いろい

ろな種類が考案されている。

一番有名なのは、大学的大型コンピュータなどに接続されているドラム型のラインプリンタだろう。

ドラムラインプリンタ

図2に基本構造を示す。1本の長いドラムの表面に、丸い方向にASCII文字(128文字)の活字が刻まれている。横方向には、同じ活字が印字幅だけ並んでいる。

このドラムが高速で回転するわけであるが、1回転するうちに必ず印字したい文字が回ってくる。そのタイミングに合わせてハンマで叩くなどして、活字の形を用紙に転写する。この作業がドラムの一回転内に、1行の印字桁の文だけ(大型シリアルプリンタでは136桁)並行して行われる。つまり、1行がほぼ同時に印字できる。このようなものを、ラインプリンタと呼ぶのである(現在ではあまり使われていない)。

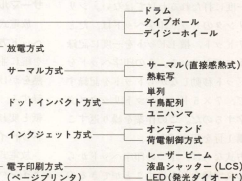
このラインプリンタは、印字ユニットを印字幅だけ並べた強引な設計だから、印字は速いが、メカニズムは大きいし価格もそれなりだし、何しろ騒音が激しい。大学や企業ではともかく、小さなオフィスや個人で使うのには適さない。

そこで、タイプライターのように1行を

●図1 印字方式の分類

活字方式

ドットマトリックス方式



1文字単位で打っていくシリアルプリンタが登場する。現在のプリンタはほとんどがシリアルプリンタだが、その印字方式は非常に種類が多い。だが、活字方式に限ればタイプボール方式とデジタイザール方式の2つが有名。いずれも英文タイプライターで使われている方式なので、実際に見て知っている人も多いだろう。

タイプボール方式

タイプボール方式は、直径30mmほどの金属のボールの表面に活字が刻まれている(図3参照)。必要な文字は、ボールを中心軸と上下方向の2方向の回転させることで選ぶ。印字はボール全体を用紙に向かってぶつけることで行う。

デジタイザール方式

デジタイザール方式は、タイプボール方式に比べれば、かなり新しい印字方式である(印字例B)。デジタイザール方式の印字ヘッド部を図4に示す。円周状に並んだ腕の先に活字が作られている。この円形の活字ユニットをステッピングモーターで回転させ、打つべき活字を一番上にもってくる。次に電磁石でハンマを突出させ活字を打ち出し、文字を印字する。タイプボール方式よりは機構が単純。

活字方式のシリアルプリンタは文字は美しいのだが、印字速度の遅さと、グラフィックスができないこと、小型、低価格化が困難なことが短所だ。

ドットマトリックス方式

活字方式の問題を解決するため、文字の形を数十個の点の集合で表現するドットマトリックス方式が考案された。

ドットマトリックスによる文字を図5に示す。この図では、1文字が縦7×横5ドットで構成されているが、この合計35のドットが一度に打たれるわけではない。シリアルプリンタのドット印字ヘッドは、たとえば縦7ドット、横1ドットを一度に記録するようにできている。この印字ヘッドを横に5ドット移動しながらドットを記録することで、7×5ドットのマトリックス文字を印字するのだ。この作業を繰り返すことで、横1行を印字できる。

この方式ならば、1つの印字ヘッドからすべての文字種を打ち出すことができるので小型軽量、低コスト化が可能。また、グ

Type Style: Courier 10

IBM Electronic Typewrit
applications, using int
choose the style of typ

●印字例A プリントホール方式のIBM「セレクトリックタイプライター」の印字例。タイプボール方式に近似している

abcdefghijklmnopqrstuvwxyzhijklm
ABCDEFGHIJKLMNPNP
!@#\$%^&*~`|_<>(){}[]
.?Ω°AEC AAA AAAAAAA

●印字例B デジタイザール方式のタイプライター「AURORA 113」の印字例

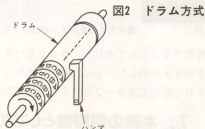


図3 タイプボール方式

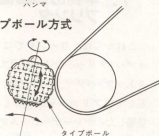


図4 デジタイザール方式

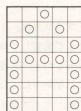
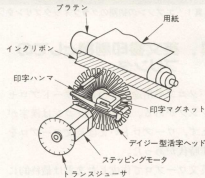


図5
ドットマトリックス
方式による文字

ラフィックスへの対応も容易だ。

またドットマトリックス方式にも数々の種類があるが、マイコン用のプリンタとして初期に登場した放電方式から説明しよう。

放電方式

特殊な記録紙の表面に接した電極に高電圧をかけて、記録紙の表層を溶かすことでドットを記録する方式である(図6)。

この方式の特徴は、何といっても安価なこと、1万円台の製品があった(松下電機「EUY-10E」)。記録紙自体が変色するため、リボンも不要である。ただ、記録紙は艶のない銀色をしており、見た目が悪く文字も読みにくかった。

サーマル方式

放電式の欠点を改良したのが、サーマルヘッド方式(感熱方式)だ。小型の発熱体を縦に並べた印字ヘッド(図7)を使い、熱をかけると変色する感熱紙にドットを記録する。感熱紙は色が白で、ほとんど普通紙と変わらず、文字も読みやすい。

ドットインパクト方式

放電式、サーマル方式と並行して、ドットインパクト方式のプリンタも研究されてきた。放電、サーマルが用紙に化学的な変

図6 放電方式

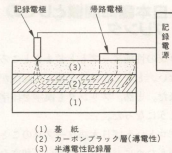


図7 サーマルヘッド方式

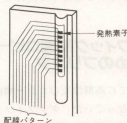


図8 ドットインパクト方式の印字部



化を起こさせてドットを記録するのに対して、ドットインパクトでは印字ヘッドから細かいピンを突出させ、インクリボンを用紙に打ちつけてドットを記録する(図8)。

放電方式、サーマル方式が専用記録紙を必要とするのに対して、普通紙に印字できるというメリットがあり、その後普及することとなった。ただし、メカニズムが複雑なため、初めは非常に高価であった。

この方式は、米国のセントロニクス社が基本構造を考案したものの、それを日本のブラザー工業が製造し「M-101」写真3参照)、その後信州精工や精工舎などが参入して、低価格化、高性能化競争が始まった。信州精工はエプソンと社名変更し、いまや押しも押されぬプリンターメーカーである。

低価格ドットインパクトプリンタのパイオニアは、精工舎である。独自のユニハンマ方式を開発し、それまで10数万円していた80桁プリンタを6万円台で出した(「GP-80」)のである。ユニハンマ方式(印字例C)は、1本のハンマによって縦7ドットを印字できるシンプルなメカニズムだった。その秘密は用紙の裏で回転している突起付きのプラテンにある。突起が打つべきドットの位置に回ってきたタイミングに合わせて、ハンマ前からハンマで打つのだ(図9)。

印字品質の改良

ドットマトリックスプリンタでは、初めは7×5ドットで文字を構成していたが、印字品質を向上させるため、9ドットのヘッドを持った機種が増えてきた。文字の縦方向のドット数は、ヘッドのドット数で決まってしまうが、横方向にはヘッドの移動間隔を小さくすることでドット数を多くできる。標準文字の横ドット間隔の半分のピッチにした倍密度、4分の1にした4倍密度などグラフィックスを印字できるグラフィックプリンタも登場した。

漢字印字のためには、文字を少なくとも16×16ドットで構成する必要がある。これはそれまでのドットマトリックス方式に比べて、縦に倍の密度で印字することになるが、印字ピンをそのように狭い間隔で配置することは困難だった。そこで、印字ピンを横2列に半ドットずらした千鳥配列のヘッドが考案された(図10)。

16ドットのヘッドはコスト高だったため、

図9 ユニハンマ方式

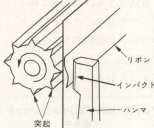


図10 24ピン千鳥配列→



●印字例C 精工舎のユニハンマ方式によるドットインパクト方式のプリンタ「GP-80」の印字例

```
! " # $ % & ' ( ) * + , - . / : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` { | } ~ ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

●印字例D プリンタ内蔵のワープロ、56ドットの熱転写ヘッド採用、「キヤノワード400」の印字例(キヤノン)

大文字	56ドット	12P
-----	-------	-----

画面を見ながら、ごく自然に、思いのままにプレイ操作の手間も時間も省き、入力スピードが大幅にアップ完成しました。

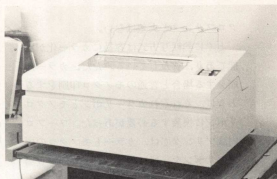
●9ピンヘッドを使って1行の往復で縦18ドットの漢字を印字するものもあった。
現在は24ドットヘッドが主流となっている。

インパクト方式はメカニズムが複雑で、とくに24ドットプリンタは高価だ。そこで、安価ながら高品質な印字ができる**熱転写方式**が台頭してきた。

熱転写方式

先に述べたサーマル方式を改良したこの方式は、サーマルヘッドによりフィルムリボンに塗られたインクを溶かして用紙に転写するものだ。サーマル方式に比べて、普通紙に印字できるメリットがある。また、ヘッドに稼動部がないためコンパクトでコストもかからない。消費電力が少なからず電源が小さく、ここでもコストとコンパクトさで有利になる。そしてインパクト方式のような騒音も出さない。

現在ではワープロ用に56ドットというものまで作られている(キヤノン「キヤノワード400」など)いまや印字品質の点で、熱転写方式はレーザービーム方式以上なので



●写真3 日本で最初にドットインパクト方式を採用した、ブラザー工業の「M-101」

ある(印字例D)。

カラー化

印刷物の場合、すべての色は色材の3原色であるシアン、マゼンタ、イエローの組み合わせで表現できる。3色をすべて混ぜれば黒になるが、きれいな黒を出すため、黒の部分には黒インクを使う。

プリンタの場合も4色のカラーリボンを使えば、カラー化ができる。

シャープが4色カラーのインクジェットプリンタを出したことがあるが、高価でメンテナンスに少し難があった。

初めてパーソナルなカラープリンタを出したのは精工舎。同社独自のユニハンマ方式をうまく利用して、4つのハンマでそれぞれシアン、マゼンタ、イエロー、ブラックの4色を打ち出すプリンタを作った。だが、ユニハンマ方式は印字速度が速くないため、その後この方式は消えてしまった。

現在では、カラープリンタはごく当たり前のものになってしまい、価格的にもモノ

クロブリタと大差はない。

とくに熱転写方式は容易にカラー化が可能であり、普及している。ただ、カラー印刷をする場合と普通のモノクロ印刷をするときに、それぞれカラーリボンとモノクロリボンに換装する必要があった。ワープロ内蔵プリンタでは、カラーとモノクロの両方のリボンを一緒に載せられるものが登場している。パソコン用のカラープリンタも、今後この方向に進むだろう。

高速性の追究

ドットインパクト方式は、初期においては30CPS程度だったが、これも徐々に改善され、いまでは普及タイプでも100CPS、速いものだと300CPSを超えている。とくに、印字ヘッドの往復の動きを利用した双方向最短距離印字が登場してから、印字効率は非常にようになった。

熱転写方式は、もともとドットインパクト方式より遅く、またリボン巻き上げの関係で双方向印字ができない。それでも、初期のものよりは格段に速いものになってきているので、まだまだ期待できる。

ドットインパクト方式は、一応高速であるが印字は物理動作によるため、やはり限界がある。その限界を超えるため、新しい印字方式のプリンタが登場してきた。

1つはインクジェットプリンタである。ノズルからインクの粒子を噴出させ、用紙上にドットを記録する。方式として、オンデマンド方式と荷電制御方式がある。

オンデマンド方式

インパクト方式のピン代わりにノズルを並べ、紙にインクの粒を吹きつけて印字する方式。代表的なものにエプソンの「IP-130K」がある（印字例E）。図11にその構造を示す。インクを噴射するしくみには、圧電素子に電圧をかけると圧力が生じる原理を応用している。

荷電制御方式

図12に構造を示す。

1本のノズルから超音波振動子によりインク粒子が常に連続的に噴出されている。インク粒子を帯電させておき、偏向電極により粒子を上下に振り、縦1列印字する。印字に利用しなかった余分なインクはガーターに回収される。

オンデマンド方式よりも、高密度で印字ができ、シアン、マゼンタ、イエロー、ブラックの4色のヘッドを装備することで、簡単にカラー化が可能。このメリットを生かしたインクジェットのカラー画面プリン

タが最近登場した NEC「PC-PR801」（写真4と印字例F）。

インクジェット方式は、初期においてはインクの目詰まりなどのトラブルがあったが、現在ではそれを克服する技術が確立し

図11 オンデマンド式(ピエゾ式)インクジェット

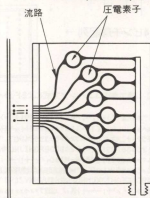
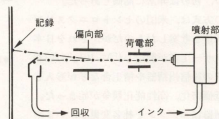


図12 荷電制御方式のインクジェット



●印字例E オンデマンド方式の代表、エプソンの「IP-130K」の印字例

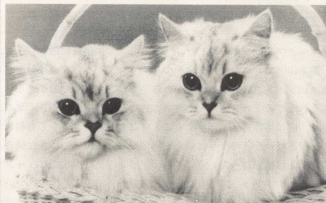
新時代のインクジェットプリンタ。
エプソンのニューテックノロジーがまた一つ結実。
これまでにない静かな環境の実現。
高印字速度・高印字品質を可能にしています。
IP-130Kはオンデマンド型インクジェット方式を採用。
オフィスに最適な静かで高速なプリンタです。
本格的ビジネスユースからパーソナルユースまで幅広いニーズに応える、
まさに漢字プリンタ熟成時代の倒来をつける多機能プリンタの登場です。



●写真4
NECのカラービデオプリンタ「PC-PR801」

●印字例F
「PC-PR801」の
印字見本例

NEC PC-PR801



フルカラーイメージプリンタ

ている。

さらに印刷速度を追究して、たどり着いたのがページプリンタ。代表的なものにNECの「PC-PR602」(印字例G)がある。

電子コピー機の印刷メカニズムを応用し、原稿のコピーの代わりに文字を印刷できるようにしたものだ。コピー機のように1ページ単位で印刷されるため、ページプリンタと呼ばれる。

静電ドラム上に形成された潜像にトナー（黒粉）を付着させ、紙に転写、熱で定着させる機構はコピー機そのもの。図形の潜像をドラム上に形成する方式に、レーザービーム方式、LCS（液晶）方式、LED（発光ダイオード）方式などがある。

主流はレーザービームである。多角形の回転鏡によりレーザービームを静電ドラム上にスキャンさせ、文字や画像の潜像を焼きつける(図13)。

LCS方式は、ラインドットの透過型液晶シャッターで画像を形成し、光をあてて静電ドラムに焼きつける。

LED方式では、横一列に数千も並べた小さな発光ダイオードのアレイを発光させることで画像を作る。

ページプリンタは高いドット密度を容易に実現できるため文字品質がよく、32ドットや48ドットの漢字を印字できる。

速度も1ページ当たり10秒以内で印刷できるほど速く、動作音も非常に静か。

レーザービームによるページプリンタは、大型で非常に高価なものは以前からあったが、パソコンにつなげるコンパクトなものが登場したのはここ2～3年のことだ(写真5)。ところが、このわずかな期間の間に、価格は100数十万円から30万円台まで下がってしまった。実に驚くべきことである。

カラー化もすでに進みつつある。キヤノンのカラーレーザーコピー（写真6）は、まだ数百万円で基本的にはカラーコピー機だが、その気になればコンピュータとのインタフェースはすぐにも可能はずだ。あとはコストダウンと小型化を待つのみである。

プリンタの歴史を見てきたが、パソコン
なりに性能アップのスピードが速かったこ
とに気がつく。ただ、まったく新しい印字
方式が出てくることはもうないかもしれな
い。次世代は、レーザービームプリンタが
主流となるだろう。

NEC

**日本語ページプリンタ

日本語ページプリンタ PC-PR602 は、

従来のシリアルプリンタが持つ機能はもちろん、
さまざまな新機能を搭載。

これまでに無いまったく新しい表現を可能とする
次世代のプリンタです



88TERMを 使いこなす

通信は面白い

ピーッという音とともに自分の世界が広がる。たった今、自分の部屋でたった1台のパソコンで作ったデータが何万台のパソコンにつながる。アメリカにドイツにフランスに……と場所も時間も関係なく、人種も年齢も、ただコンピュータと電話回線があれば。それがパソコン通信です。

まず、パソコン通信の中でも最も親しみやすくコミュニケーションを行えるのが、電子メールです。訳してしまえば「電子手紙」ですが、考え方とすれば、私書箱で個人から個人のメールボックスにメッセージを送るものです。

郵便だと手間や時間がかかり過ぎるし、ファックスにしても、場所が限られます。となればパソコン通信が最適というわけです。見たいときに見たい場所（パソコンが電話回線と接続してなければいけません）で見てられ、文章ですだから余分な言葉がなく必要事項だけという時間の短縮ができ、データとしても残るので、ほかの人に送ることもできます。

電子メールは、受信者を指定するのでそこから必然的に共有範囲を狭めているわけですが、電子掲示板や電子会議室では、多くのコミュニケーションがとれ、SIGでは、同じ趣味をもつ人々と意見が述べあえるのと同じです。

また、瞬時のコミュニケーションの送受信といったら、チャットが一番にあげられるでしょう。チャットとは複数の会員同士

がリアルタイムで筆話することです。

パソコン通信は単に意見を述べあう場所ではないのです。百貨店ではオンラインでカタログ注文を受けており、閉店時間も気にせず、テレフォンショッピングのように話し中ということなく24時間体制です。

またオンラインゲームなどは自分の作ったゲームをオンラインで競うのですから、これこそコンピュータゲームでスリル満点ではないでしょうか。パソコン通信の中でも最も重視されているのがデータベース。莫大な量の情報を蓄積したもので、調べたいことのキーワードをいくつか入れていけば、検索し、データが出てくるというコンピュータならではの機能を利用したものです。新聞の速報や記事も入っていて、情報の入手は速く、図書館のように資料を探す手間もなく、自分の得たいデータが瞬時に得られます。

SOFTWAREは もりたくさん

パソコン通信が盛んになるにつれ通信ソフトも色々なものが発売され、操作がカーソルとRETURNキーで扱える簡単なものから、グラフィックスが楽しいものや、アクセサリがたくさんついているものと多岐に販売されています。

各ネットごとに通信条件の設定ができたりオートダイヤル（自動的にモデムが電話をかける）、オートログイン（自動的にID入力などをしNETに入る）を行えるのですから、ほとんどの人がパソコンのターミナルモードを使っている通信より市販の通信

通信は98ユーザーだけのものではありません。98ユーザーだってちゃんとモデムもソフトもあるのです。ネットワークがこれだけ普及してくれば利用しない手はないのです。そこで、ここではPC-8800シリーズ対応の通信ソフト「88TERM」で実際にアクセスしてみます。また、本誌152ページからモデムのカatalogも参照してください。

ソフトを目的に合わせて使い、楽しんでいます。

通信用ソフトは色々なメーカーから多くの種類が発売されているので自分にあった価格、機能を自由に選ぶことができます。PC-8800シリーズ用には通信ソフトとして基本的な機能がそろい低価な「PLAY NET」(工画堂スタジオ 6,800円)、ワープロ「JET」と一緒にになっているもので、初心者でも使いやすい「JETターミナルペディア」(キャリアラボ 22,000円)、PC-8800シリーズ全機種に対応の「88TERM」(アスキー 12,800円)などがあります。

今回ご紹介するのは、PC-8800シリーズ用の通信ソフトで多く使われている「88TERM」です。

「88TERM」試運転

パソコン通信には特別な技術や難しい知識は必要ではないと述べてきましたが、セッティングもそのとおりです。電話、モデム、パソコンをつなげばよいだけなのです。たった3本の線で結ばれている簡単なセッティングです。

「88TERM」は基本動作をメニュー方式で選択でき、初心者でも使いやすくなっています。また、マニュアルも2冊のうちの1冊はマンガで解説され、初心者にも分かりやすいようになっています。

まず、ソフトを立ち上げるに「ASCII」文字が赤や青にきれいに出てきます(写真1)。通信時や設定時の画面は至ってシンプルで使いやすくなっています。

写真 1

88TERMの初期画面

ASCII

写真 2

ネットワーク情報設定画面でログインするネットワークを選ぶ

```

Network name      Tel. number
-----
04 15 23pm 00 01 00

```

写真 3

通信条件設定の画面

```

Network name is PC-VAN
Telephone number is 045-681-1476
Auto login is ON
Speed is 1200 baud
Parity bit is 8 Data none
Stop bit is 2
Listen mode is Full with Echo
Response to XON/XOFF is ON
Generate XON/XOFF is ON
CTS Control is OFF
Serial terminator is CR
Receive terminator is CR LF
Serial code is RS 422
Emulation is PC801
Send speed is 9600
Printer mode is Parallel printer
Parity is 8
Mirroring is OFF
Overwrite is OFF

```

04 16 16pm 00 00 00

通信ソフトがあるから、すぐ通信できるのではないのです。各ネットワークにより通信条件が違うわけですから、それを設定しなければなりません。それをネットワーク情報の設定といいます、これをしなかったり正しい設定を行わなければ、通信できませんので重要です。これを、最初に設定します(写真2)。ネットワーク情報には3種類あり、まずネットワーク名称です。

HELPキーを押すとコマンドメニューが表示されるので、カーソルを動かし「Modify」に合わせます。RETURNキーを押すとModifyサブメニューが出るので、「Com. Setting」にカーソルを合わせます。ネットワーク情報設定画面が表示されるので名前、電話番号を入れ、オートダイヤル(自動的に電話をかける機能)の設定をします。

これで、どのネットワークに接続するかを選ぶときに出てきますので、同じネットワークでも違うアクセスポイントを入力するときなどは、細かく書き込むと分かりやすく便利です。通信手順の設定も各項目をカーソルを動かして選んで設定できるので、通信速度やビット長、ストップビット長、XON/XOFFなど、通信する相手との通信条件の設定をします(写真3)。

そしてもう1つはターミナル情報ですが、これはPC-8801側の設定で、エミュレーションモードや、画面に表示する文字を決める画面モードの設定です。

88TERMはオートダイヤル機能だけでなくオートログイン機能(自動的にID番号やパスワードを入力する機能)もあるので、ID番号やパスワードなど、ネットワークにログインしたときに決まって入力する文字を設定しておくことができます。

ですからソフトを立ち上げてネットワークを選択してRETURNキーを押せば、もうネットワークに入れてしまうのです。

PC-VANにアクセスしてみよう

さていよいよネットワークに接続です。日本最大の会員数を誇るネットワークNECの「PC-VAN」に入ることにしましょう。アクセスポイントは全国86か所、SIGも80以上という幅広い年齢層とテーマを誇っ

ています。

通信先に PC-VAN を選び RETURN キーを押すと、モデムの音が鳴り始めます。「ジョーコ」(電話をかける音)「ツルルツルル」(電話のコール音)「ビー」(接続した音)「ガー」(なんだか分からないけど恐い)アッというまにドンドン進んでいきます。でも、ここで焦ってはいけないのだと思っでも、ここでも最初は何が起るか分からないのでドキドキもんです。

WELCOME TO C&C-VAN

このメッセージから PC-VAN は始まり

ます。そして自分の世界がドンドン広がっていきます。

ID 番号、パスワードと入力されていきます(オートログインではない人はここでとくに焦ってはいけません。間違えるとハッカー防止機能のため「See You Again」となりますので)。最初の通信時にプロフィールの設定をします(次回からは変更がなければ登録できるので必要ありません)。通信ソフトでさんざん条件設定をしたのに、通信してまで設定かと思うでしょうけど、ネットワークの全員には色々なコンピュータを持ってる人がいるので、漢字が使えない人から使える人と様々で、その機種にあった設定が必要になります。画面サイズや漢

字 JIS、コマンド列の表示などの設定が通信の第一歩ですので、間違えるとほかの人がメッセージを読めなくなったりします。また、ここでの設定で漢字が読める人か読めない人が電子メールを送ったときに表示されます。

PC-VAN のメニューに入ると 13 に分かれていて(写真 4)、ニュース/スポーツや SIG、ショッピングなど、見たいものを選びます。13 のメニューもそれぞれ選ぶと個々にまた分かれているので(写真 5)、全部のメニューを見ることは不可能ではないかと思うほどです。「88TERM」はアップロード機能があるので、ほかのワープロソフトであらかじめ作っておいた文章(お料理番組の要領ですね)を送信時に送ることができ、漢字変換機能もあるのでリアルタイムに仮名や漢字で送信できます。またダウンロード機能もあるので電子メールや SIG の内容、その他のデータがあつてびっくり見たいときは、ディスクに落としてあとでワープロソフトで読んだり、プリントアウトしたりもできます。

それらの機能をカーソルを動かすことによつて簡単に行えます。

フロッピー内のファイル、内蔵時計を管理して、時間が常に画面の右下に表示されていますので、通信時刻がすぐに分かり、電話料金や通信料金を心配しながら通信するということがなく心ゆくまでネットワークワーキングが楽しめるのです。

総合評価

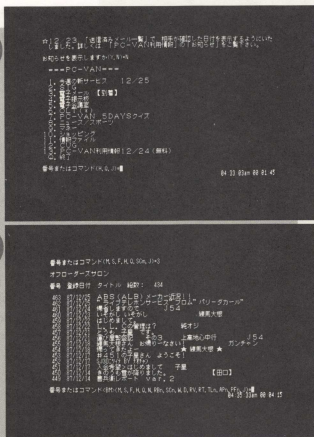
以上のように「88TERM」は PC-8800 シリーズ用の通信ソフトとしては、全体的にバランスがとれ、PC-9800 シリーズ用のソフトに比べてもあまり見劣りしません。エディタが付属していないのが少々寂しい気もしますが、88 の処理速度を考えると付属していないのが、かえってよいのではないかと思います。しかし、一番の難をいえば、筆者のようなアナログの頭をもった女の子にはシンプルな画面が少々分かりずらい気もします。アクセサリがついていて色とりどりのほうが楽しいと思うのはワガママなんでしょうか?

写真 4

「PC-VAN」メニュー画面

写真 5

メニューのタイトル別画面



PC-8800シリーズ全機種対応

MS-DOSの環境をディスクBASICで

N88-BASIC

COSMOS

みなさんは「IDOS88」をご存知でしょうか。88上でのN88-BASICのファイル操作環境を16ビット並みにしてしまうすごいツールです。しかし、本誌の読者の方には、まだまだ通常のディスク BASIC のほうが慣れてて使いやすく、手放せない、という向きも多いことでしょう。このプログラムはそんな方たちのために、通常のディスク BASIC の動作環境を MS-DOS に近づけることを目的に作られた画期的なプログラムです。

目標はMS-DOS!

COSMOSの構想

いま、88の動向の1つとして、「何とかして98に近づこう」とするものがあります。ハードウェアではPC-88VAの登場、ソフトウェアでは「IDOS88」の登場がその表れであるといえるでしょう。

また、「IDOS Tools #1」により、いよいよ IDOS 上で動作する BASIC インタプリタがリリースされようとしています。これでは、98と並ぶレベルに達するわけです。さて、IDOS 上で BASIC が動くようになったとします。しかし、それでも通常のディスク BASIC を使いたいという場合があります。それは、

IDOS、あるいは MS-DOS を使い慣れて、ちょいとディスク BASIC を動かしたときのファイル操作の不便なことといったことはありません。BASIC が IDOS のもとでグレードアップしたとしても、通常のディスク BASIC はまだ少しも成長していないのです。

通常のディスク BASIC の操作環境も、何とかして MS-DOS に近づけることができたいものでしょうか。

このような構想で COSMOS が作られました。COSMOS は、通常のディスク BASIC 上で動作し、その操作環境を MS-DOS に近づけることを目的とするシステムです。

目標は MS-DOS です。MS-DOS の仕様を復習しながら、COSMOS の仕様を構築していくことにしましょう。

コマンドインタフェースの役割=スクリーンエディタ

MS-DOS、IDOS において、実際に DOS の機能を使用するためのコマンドを受けつけるのは、コマンドインタフェースと呼ばれるプログラムです。このプログラムは DOS の土台からは独立しており、1つのコマンドファイルとしてシステムディスクに用意されているものです。独立したファイルとして用意されているため、バージョンアップした新しいコマンドインタフェースと容易に交換することができるわけです。

ディスク BASIC でコマンドインタフェースに相当するのは、ROM に内蔵されているスクリーンエディタです。このスクリーンエディタに、これから述べる数々の機能を追加するために、増設部分を作ることになります。

その増設部分を1つのファイルとしてディスク上に用意します。ディスク BASIC そのものを作り変えることなく、ディスク BASIC をグレードアップするわけです。

この増設プログラムがリスト1の「COSMOS」です。このプログラム COSMOS に次の注文をつけることにします。

まず、すべての N88-BASIC Disk version の上で動作すること。

日本語 BASIC 上でも動くことが必要。次に、

できるだけコンパクトに作られること。

既存の BASIC プログラムが従来と変わらなく動作するためには、フリーエリアの減り方を最小限に抑えなければなりません。

さらに、以下に述べる機能を持たせることにします。

1 内部コマンドと外部コマンド

MS-DOS、IDOS では、コマンドインタフェース自身がいくつかのコマンドを実行する機能を持っています。このコマンドを内部コマンドと呼びます。そのほかに、ディスク上に用意されたコマンドファイルがあります。コマンド1つにつき、1つのファイルが対応しており、コマンドインタフェースが受けつけたコマンドが内部コマンドでないとき、コマンドと同名のファイルがディスクからロードされ実行されます。これが外部コマンドです。

外部コマンドというのは、強力な構想です。システムの機能をあとからいくらでも増設することができるわけですから。

ディスク BASIC ではコマンドはすべて内部コマンドであるといえます。COSMOS には、BASIC の命令に新しく数種の内部コマンドを増設し、外部コマンドの実行機能を与えます。外部コマンドは今後いくつでも追加して作っていくことができます。

今回は、もっとたくさん外部コマンドを作って、システムとしてそれなりのものにしたかったのですが、時間がなくて2個にとどめました(リスト2とリスト3)。

2 デバイスファイルとI/Oリダイレクト

キーボード、スクリーン、プリンタ、RS-232C 通信回線といった入出力装置を、ディスクのファイルと同様に扱う構想は N88-BASIC ではすでに実現されています。

たとえば、“SCRN:”という名前でファイルを開き、このファイルに PRINT # や PUT # でデータを出力すると、通常の PRINT 文を実行したのと同様にスクリーンにデータが出力されます。ファイル名を“LPT:”とすれば、プリンタに印字されます。このように、デバイスごとに特別な命令を用意することなく、ファイルとして扱うことで命令のしかたをすっきりさせたのがデバイスファイルの考え方です。

MS-DOS、IDOS ではさらに、標準入出力という考え方が採用されています。あたかもキーボードから文字を入力するように、ディスクのファイルからデータを入力したり、普通ならスクリーンに出力される文字をディスクのファイルに出力したりすることができます。このように、“ふつうならキーボードから入力してスクリーンに出力する内容”を入出力するファイルを標準入出力と呼びます。通常は標準入出力のファイルはキーボードとスクリーンになっていて、これをプリンタやディスクのファイルに変更することができるのです。この操作を I/O リダイレクトと呼びます。

COSMOS には、この I/O リダイレクトの機能を持たせることにします。

3 バッチ処理とテンプレート

MS-DOS、IDOS では、コマンドをいくつもまとめて実行することをバッチ処理と呼びます。COSMOS では、BASIC の機能がそっくり使えますから、コマンドをプログラムにして RUN すればそれがバッチ処理となります。

テンプレートというのは、要するにコマンド行の入力用のエディタのことです。BASIC はコマンド待ちの状態がフルスクリーンエディタであり、また HELP キーを押すと直前に入力したコマンド行の内容がそっくり表示されるようになっていきますか

ら、とくに考えることはありません。

COSMOSに搭載しない機能

ツリーディレクトリ、ワイルドカード、フィルタ、パイプといった機能は、COSMOS Ver 1.0 には与えられません。これらの機能がないのは残念なことですが、プログラムをコンパクトにし、なるべく N88-BASIC 自体に手を加えず互換性を保つためにも、これらは搭載しないことにします。

外部コマンドの柔軟性

MS-DOS では、外部コマンドがほかの外部コマンドをロードして実行するなど、いくつものプログラムファイルを同時に取り込んで実行することができます。これは外部コマンドをロードするアドレスが特定されておらず、どこにロードしても実行できるという16ビット CPU の特性を生かした機能です。

COSMOS には、この機能を搭載します。外部コマンドのプログラムファイルはすべてリロケータブルに加工し、メモリ上のどこにロードしても実行できるようにします。これにより、外部コマンドがほかのコマンドファイルをロードして実行することができます。

しかし、そういう加工ができるようなプログラムは、それができるように注意しながら作らなければなりません。ユーザーが自分で外部コマンドを作るときに、あまりやっかいな制約がないような、コマンドファイルの形式を定めなくてはなりません。

BIOSファンクションコール

MS-DOS、IDOS では、ファイルやデバイスの入出力などをひとまとめのサブルーチンにしています。そうすることで、あとから外部コマンドを作るときに入出力が簡単明瞭に行えるようになっています。

残念ながら COSMOS ではこの機能も搭載しないことにします。入出力などは直接 ROM 内ルーチンをコールして行います。しかし、ROM 内ルーチンにはない、COSMOS に特有ないくつかの処理については、外部コマンドから呼び出して使えるように

考慮しています。

COSMOSが動作する環境

COSMOS は、88のすべての V1/V2 BASIC 上で動くことを目標にしています。プリンタや RS-232C 通信回線はもともと N88-BASIC がサポートしていますから対応可能です。ディスクは 1D、2D、2DD、2HD、8インチ 2D、日本語 BASIC に対応します。

COSMOS のディスクファイルは、システム本体が1つと、その他の外部コマンドファイルで構成し、ディスクコードにはいっさい手を加えません。むしろそれによって、改造されたディスク BASIC 上でも動作するようにしています。

COSMOS は「BLOAD, R」で起動することにします。ディスク BASIC のユーティリティを使って ID セクタを書き換えれば、COSMOS のオートスタートが可能です。

リストの打ち込み方法

- ① N88-BASIC V1/V2モードのディスク BASIC を起動し、

How many files (0-15)?

には RETURN キーを押してください

- ② CLEAR、&HBFFF ☺

MON ☺

を実行してください

- ③ リスト1を打ち込むときは、

EC900 ☺

リスト2, 3を打ち込むときは、

EC000 ☺

と入力してください。

- ④ リストを入力してください。ROLL UP, DOWN キーで画面をスクロールさせることができます

- ⑤ 打ち込みが終わったら、ESC キー、CTRL-B キーを押して BASIC にもどります

- ⑥ リスト1をセーブするときは、
BSAVE "COSMOS", &HC900,
&HA2C ☺

リスト2をセーブするときは、
BSAVE "MKCOM.COM", &HC000,
&H444 ☺

リスト3をセーブするときは、
BSAVE "FCOPY.COM", &HC000,

を実行してください

リスト2とリスト3のファイル名には必ず大文字を使ってください

- ⑦リストAの統一チェックサムダンププログラムをロードして実行してください。リスト1, 2, 3のチェックサムと比較して、打ち込みミスがないか確認してください

COSMOSの 起動方法

N88-BASIC V1/V2モードのディスクBASICを起動します。日本語BASICでもかまいません。

How many files (0-15)?
には必要な数を入力してください。

BLOAD"COSMOS", R
でCOSMOSが起動します。このステートメントを、ディスクBASICの起動と同時に自動実行すれば、COSMOSがオートスタートします。それにはBASICシステムディスクのユーティリティを使用してください。

COSMOSを起動したとき、BASICの変数はクリアされます。

タートルグラフィック拡張命令もCOSMOSと同時に使用できます。

コマンドの使い方

●STDIN (内部コマンド)

[機能]標準入力をリダイレクトする

[書式]STDIN<ファイルディスクリプタ>

通常キーボードから入力すべきデータをファイルから入力することを宣言します。ファイルとしてはディスクのファイルのほかに、"COM:" (通信回線) が使用できます。カセットテープからの入力は不可です。

STDIN"KYBD:"

あるいは単に、

STDIN

とすると、標準入力はキーボードにもどります。

また、「OK」が表示されたときにも標準入力はキーボードにもどります。ダイレクトステートメントで標準入力をリダイレクトしたまま、何らかの処理を行いたいときには、STDIN文と目的の処理実行文とを

続けてマルチステートメントにしてください。

そしてファイルをすべて入力し終わったときにも、標準入力はキーボードにもどります。

ディスクのファイルを標準入力とする際には、ファイルの内容はいったんすべてテキストRAMの空き領域にロードされます。このため、標準入力をリダイレクトしたまま、大きなBASICプログラムをロードすると、標準入力のデータが失われる可能性があります。データが失われてしまったときに限り、標準入力はキーボードにもどります。

標準入力がリダイレクトされた状態では、キーボードからの入力はSTOPキーとESCキーのみが有効です。STOPキーを押すと標準入力はキーボードにもどり、STOPキーの処理が行われます。ESCキーは、押している状態で動作を停止し、離すと動作を再開します。

例 STDIN"KEYS":RUN"PROG" ◎

とすると、ディスク上のBASICプログラム"PROG"をロードして実行を始めますが、プログラム中のINPUT文、INKEYS、INPUTsなどでは、キーボードからではなく、ディスク上のファイル"KEYS"からキャラクタを読み込んで処理します。

●STDOUT (内部コマンド)

[機能]標準出力をリダイレクトする

[書式]STDOUT<ファイルディスクリプタ>

通常スクリーンに出力すべきデータを、スクリーンに出力すると同時にファイルへも出力することを宣言します。ファイルとしてはディスクのファイルのほかに、"COM:" (通信回線)、"LPT:" (プリンタ) が使用できます。カセットテープへの出力はできません。

STDOUT"SCRN:"

あるいは単に、

STDOUT

で、標準出力はスクリーンにもどります。

また、「OK」が表示されたときにも標準出力はスクリーンにもどります。ダイレクトステートメントで標準出力をリダイレクトしたまま、何らかの処理を行いたいときには、STDOUT文と目的の処理実行文と

を続けてマルチステートメントにしてください。

標準出力をリダイレクトした状態では、スクリーンへの出力が同時に行われます。

すべてのファイルをクローズしてしまうようなBASICステートメント、たとえばCLOSE、CLEAR、RUN、LOADなどが実行されると、標準出力はスクリーンにもどります。また、標準出力をリダイレクトしたまま、BASICファイルをロードして実行したいときには、CHAIN文を使用してください。

例 STDOUT"DISP":CHAIN"PROG" ◎

とすると、ディスク上のBASICプログラム"PROG"をロードして実行を始めますが、実行中にスクリーンに表示される文字はすべてディスク上のファイル"DISP"に記録されます。

STDOUT"..." : FILES 2 ◎

のように、異なるディスクのファイル出力をディスク上のファイルに標準出力すると、正しい出力が得られず、エラーが出ることもあります。

●FDD (内部コマンド)

[機能]5インチディスクの種類を設定する

[書式]FDD [<ドライブ番号>]

[, <ディスクモード>]

1D, 2D, 2DD, 2HD 5インチミニフロッピーディスク種別を設定します。ディスクモードの値には上記の順に1, 2, 3, 4を設定します。2Dドライブのマシンでは1Dと2Dすなわち1と2のみ有効です。

ドライブ番号を省略するか、0を設定すると、すべての5インチディスクユニットに対して設定を行います。

ディスクモードを省略するか、0を設定すると、

①2Dドライブのマシンでは、2Dに設定します

②2HDドライブのマシンでは、ドライブにディスクがセットされていればそのディスクの種類を自動識別して設定します (ただし、そのとき1Dディスクがセットされていても2Dに設定されます)

例 FDD ◎

FDDを2HDドライブのマシンで実行すると、5インチディスクユニットの

うち、ディスクがセットされているドライブについて、それぞれのディスクの種類を自動識別して、ディスクアクセスができるよう設定します。

●DRIVE(内部コマンド)

[機能] COSMOS外部コマンドファイルをロードするドライブ番号を変更する

[書式] DRIVE <ドライブ番号>

外部コマンドを呼び出すドライブを設定します。通常、外部コマンドは、COSMOSを起動したファイルが存在するドライブから呼び出されます。単に、

DRIVE

とすると、そのドライブにもどります。

例 DRIVE 2

この場合は、以後外部コマンドはドライブ2からロードされるようになります。

●VER(内部コマンド)

[機能] COSMOSのバージョン番号と、内部コマンドの一覧を表示する

[書式] VER

COSMOS起動時に表示されるメッセージと同じ表示を行います。内部コマンドを増設するような外部コマンドが実行されたあとでは、増設された内部コマンドも表示します。

●MKCOM(外部コマンド)

[機能] 外部コマンドのコマンドファイルを生成する

[書式] MKCOM <コマンド名>

後述の「外部コマンドの作り方」に従って作られたバイナリファイルから、コマンドファイルを作成します。

用意すべきバイナリファイルは、あとで述べるように2つです。たとえば「TEST」という外部コマンドを作るなら、「TEST」および「TEST.OFS」の2つのバイナリファイルを用意します。これらは同一のディスクにセーブしておきます。そして、

MKCOM"TEST"

を実行すると、同じディスク上に「TEST.COM」というコマンドファイルが生成します。

MKCOM文で指定するコマンド名には、拡張子も拡張子の前のビリオードもつけないけません。ドライブ番号をつけることはできます。またコマンド名、用意するバイナリファイルのファイル名は、大文字でな

ければなりません。

ここで「File not found」のエラーが出力されるときは、前記の2つのファイルが正しく用意されていないことを示します。

「Bad file format」(日本語 BASIC では「ファイルのフォーマットが正しくありません」)のメッセージが出力されるときは、後述の「外部コマンドの作り方」が守られていないことを示します。

例 MKCOM"3:TYPE"

とすると、ドライブ3に用意したバイナリファイル「3:TYPE」、「3:TYPE.OFS」からコマンドファイル「3:TYPE.COM」を生成します。

●FCOPY(外部コマンド)

[機能] ファイルの複写、あるいはデバイスへの入出力を行う

[書式] FCOPY<複写元ファイルディスクリブタ>,<複写先ファイルディスクリブタ>

ファイルをコピーします。異なる種類のディスク間でもかまいません。BASIC中間言語形式、バイナリファイル、ライトプロテクト、暗号化ファイル、リードアフターライトといった種別もそのまま複写されます。

ディスクのファイルだけでなく、「COM:」(通信回線)、「KYBD:」(キーボード)、「SCRN:」(スクリーン)、「LPT:」(プリンタ)を指定して入出力することができます。

キーボードを複写元とすると、キー入力待ちになってカーソルが点滅します。この状態からキーを打っていき、1行ずつ RETURN キーで改行します。終了時には STOP キーを押したのち RETURN キーを押します。こうすると、複写先がディスクのファイルなら、テキストファイルが作成されますし、プリンタなら入力した内容が1行ずつ印字されます。

例 FCOPY"COSMOS","2:COSMOS"

とすると、COSMOSのシステムであるバイナリファイルを、ドライブ1からドライブ2へ複写します。また、FCOPY"DISP","SCRN:"とすると、ファイル"DISP"の内容をスクリーンに表示します。あるいは FCOPY"LIST","LPT:"とすると、ファイル"LIST"の内容をプリンタに印字します。

外部コマンドの作り方

外部コマンドは、BASIC ステートメントの処理ルーチンと同等のレベルで処理を行うには、そこから外部コマンドを作る場合には ROM 内部ルーチンに関する知識が必要です。これについては、読者の要望があればまたの機会に詳しく解説することになります。市販の解説書でもすでに詳しい内容が公開されていますから、とりあえずステートメントの処理ルーチンは作ることができたとして、これを COSMOS の外部コマンドとして登録する際の制約についてだけ述べることにします。

外部コマンドは、各自手持ちのアセンブラを使って作成してください。

●外部コマンドの開始アドレス

プログラムの位置はどこでもかまいません。あとで BASIC のバイナリファイルとして BSAVE でセーブすることになりますが、そのときのアドレスとプログラム作成時のアドレスが違っていてもかまいません。アセンブラの都合のよいところにオブジェクトを作成します。

ただし、ORG 命令はソーステキスト中にただ1つとし、ORG で指定されるアドレスを任意に変更しても動くようなプログラムでなくてはなりません。

外部コマンドは、実行開始の時点ではあたたかみディスクコードの一部のようにして RAM 上に存在しています。すなわち 8400 H~E5FFH のどこかに存在しています。このとき 0~7FFFH にはメイン ROM がセレクトされています。

●リケータブルにするための制約

作成したプログラムは、あとで MKCOM コマンドを用いてリケータブルに加工します。そのための制約として、次のことに注意してください。

①プログラムが自分自身の内部のアドレスを指すような命令、たとえばプログラム内部で用意したワークエリアのアドレスを絶対アドレス2バイトで指すような場合、そのアドレスとしての数値を1バイトずつに分離したり、負の値にしたりしてはいけません(図1 A・B)

②プログラムは先頭アドレスから実行されるものとし、プログラムの先頭に図2の

ような情報を記しておくこと

●保存すべきレジスタ

BASIC テートメントの処理ルーチンと同様です。エラーを出さずこなすべく処理を終了するときには、まず自由に使えるレジスタはAF, BC, DEです。その他のレジスタを使うときはあらかじめPUSHしておきます。HLはテキストポインタで、外部コマンド起動時にはパラメータの先頭を指しています。必要なパラメータを読み込んだら、HLはステートメントの終わりを指していますから、その値を合わせてから処理を行ってください。処理がすべて終わったらその値をHLレジスタにもどしてからRETURN命令を実行して処理を終了してください。

●ほかのルーチンの呼び出し

① ROM 内ルーチンの呼び出し

0~7FFFH にはメイン ROM がセレクトされているのでそのままコールします。

② BASIC ステートメントの呼び出し

実行したいステートメントのパラメータを中間言語形式で並べ、そのテーブルの先頭アドレスを HL に入れて、

DEC HL

RST 10H

としたのち、ステートメントのエントリを
コールします

③ COSMOS 内部コマンドの呼び出し

②と同様にパラメータを並べたテーブル
のアドレスを HL に入れ、アドレス C9CH
をコールします

④ COSMOS 外部コマンドの呼び出し

あとで述べる EXTERN というルーチンを使います。HL に外部コマンドのエントリアドレスが入ってきます。

(EXTERN は、外部コマンドをロードした上、リロケート処理を行ってくれます)

RETURN 先を PUSH して、さらに HL を PUSH し、呼び出す外部コマンドに必要なパラメータのポインタを HL に入れて、

DEC HL

RST 10H

RET

とすれば、外部コマンドをコールしたことになります。

● COSMOS ワークエリアの読み出し

COSMOS は起動時のシステムの状態によって、ロードされるアドレスが変動しま

図 1 A 外部コマンドの制約

<p>悪い例 絶対アドレスをワードで記述していない</p> <pre>LD A, L ADD A, ADDR (下位8ビット) LD L, A LD A, H LD A, ADDR. R8 (上位8ビット) LD H, A : ADDR:</pre>	<p>よい例</p> <pre>PUSH DE LD DE, ADDR ADD HL, DE POP DE : ADDR:</pre>
---	---

図 1 B 外部コマンドの制約

悪い例	正しい例
絶対アドレスを負で記述している	
LD DE, -ADDR	LD DE, ADDR
ADD HL, DE	OR A
⋮	SBC HL, DE
ADDR : ↑	⋮
	ADDR : ↑

図2 外部コマンド作成時のフォーマット

図3 COSMOSシステムテーブルの内容

(E)B16)-1A) = VRTCAD	(VRTCAD) = VRTC割り込みのエンリアドレス	
(E)B16)-1B) = VERNO	アドレスVERNOからバージョン番号のキラケラデータ'1.0'が並ぶ	
(E)B16)-16) = STDIPTR	(STDIPTR) = 標準入力バッファ読み込み中の位置のアドレス	
(E)B16)-14) = EXPAN	アドレスEXPANから外部コマンドの拡張用'COM'が並ぶ	
(E)B16)-12) = COSMOSIDAD	アドレス(COSMOSIDAD)から「COSMOS」のデータが並ぶ	
(E)B16)-10) = COSMOS	= COSMOS先頭アドレス	
(E)B16)-E) = BOTTMOMD	(BOTTMOMD) = 外部コマンド切り捨て位置のアドレス	
(E)B16)-C) = INCOMAD	アドレス(INCOMAD)から内部コマンドのコマ番と処理先のテーブル	
(E)B16)-A) = DRIVENO	(DRIVENO) = 外部コマンド読み出しドライブ番号(1~8)	
(E)B16)-8) = STDINNO	(STDINNO) = 標準入力ファイル番号(1~8)	1バイト のデータ
(E)B16)-8) = STDINNO	(STDINNO+1) = 標準出力ファイル番号	
(E)B16)-8) = STDICRLF	(STDICRLF) = 標準入力ファイルのコードのLFを無視しないラダ	
(E)B16)-8) = FUNCALL	COSMOSファンクションコールのあと先アドレス	
(E)B16)-2) = 終	の漢字コード 8F43(シフトJISコード)	

(E)B16) = ディスクコードの終わりのアドレス+1, BASICラベル領域開始アドレス、アドレスAに書かれたデータを表示し、アドレスAから8バイトに書かれたデータをアドレスBと見なし、そのアドレスBに書かれたデータを表示

す。そのためワークエリアも移動します。

外部コマンドからCOSMOSのワークエリアを使用するために、図3のようなシステムテーブルが用意されています。外部コマンドでは、まず最初に(EB16)で示されるアドレスよりも前の26バイトを自分のワークエリアにコピーしてください。ここには、ワークエリアの位置を指すアドレスが並べられています。これらのデータは読み出すだけで、書き換えても効力はありません。ワークエリアを書き換えるときは、これらの指すアドレスの内容を書き換えるという少々厄介な手続きが必要です。

前述のEXTERNルーチンは、このテーブル中のFUNCCALLをコールアドレスとするルーチンです(図3)。詳しくはあとで述べます。

●内部コマンドの増設について

外部コマンドが内部コマンドを増設したり、逆に切り離したりすることも可能です。これについては、読者のみなさんの要望があればまたの機会に説明します。

●外部コマンドの生成

以上のことを考慮しながら外部コマンドのプログラムを作り終えたら、

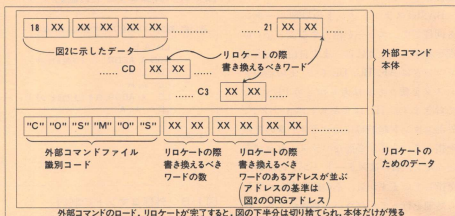
- ①大文字のコマンド名をファイル名として"BSAVE"でディスクにセーブします。ファイル名に括弧張りはつけないように
- ②ソーステキストのORG指定を下位バイトでいくらかずらし、アセンブルし直します。これを、上記のファイル名に"OFS"という括弧張りを付けたファイル名で、①と同じディスクにBSAVEします
- ③これらのファイルを外部コマンドMKCOMで処理すれば、括弧張り".COM"を持つコマンドファイルが生成します。参考のために、完成したコマンドファイルのフォーマットを図4に示します

COSMOS システムテーブル

このテーブルは、COSMOSシステムの最後にあるものですが、外部コマンドがロードされると、そのあとにコピーされます。こうして、このテーブルは常に(EB16)直前の26バイトに存在します。テーブルの内容は図3に示します。

テーブルの読み方には注意が必要です。たとえば、VRTC割り込みのエントリアドレス

図4 外部コマンド完成時のフォーマット



レスを得るためには、まずワークエリアEB16, 7に書かれたアドレスを取ってきて26を引き、そこそこの次の2バイトに書かれたアドレスを取ってきます。この値をVRTCADとします。このアドレスはCOSMOSワークエリア内を指しており、そこそこの次の2バイトに書かれたアドレスを取ってこようやくこれがVRTC割り込みのエントリアドレスです。これで、図3の読み方を理解してもらえたいでしょうか。

では、図3に示されたデータの説明をします。VRTC割り込みのエントリアドレスが、E808でないときは日本語BASICです。さらに日本語BASICでも漢字を表示できるスクリーンモード3, 4であるかどうかを知るには、アドレス8570の内容が3, 4であるかを確かめます。

標準入力力は、ディスクからの場合のみ内容をすべてメモリに読み込んでから次の処理へ移ります。(STDIPTTR)は、そのメモリの内容を読み出すポインタです(図5も参照)。テキストRAMの空き領域を使いたいときは、このアドレス以降を書き換えないよう注意します。

外部コマンドが処理を終えてRETURN命令が実行されると、外部コマンドはシステムから切断されてフリーエリアにもどります。切り捨て位置のアドレスは(BOTTOMAD)の値です。(INCOMAD)に書かれたアドレスからは、内部コマンドの文字数、コマンド名、処理先アドレスがコマンドの数だけ繰り返し書かれています。文字数として0が書かれていると、それ以上内部コマンドがないことを示し、FFが書かれて

いると、さらに次の2バイトのデータをアドレスとする内部コマンドのテーブルが離れたところから続くことを示します。

FUNCCALLは、COSMOSファンクションコールのコール先アドレスです。コールのしかたはあとで述べます。

「終」のシフトJISコードは、COSMOSがすでに起動されていることを示すためのものです。

COSMOS ファンクションコール

システムテーブルの最後から2番目のワードで示されたアドレスをコールすることによって呼び出します。処理番号をAに入れてコールしてください。

●ファンクション1 [EXTERN]

[機能] 外部コマンドをロードし、リロケートする

[entry] A ← 1

HL ← コマンド名の先頭アドレス

DE ← コマンド名の終端アドレス

+ 1

[exit] HL → コマンドファイルをロードした先頭アドレス

保存: BC, DE

すでに外部コマンドが動作中のときには、そのあとに続けてロードされます。図5のメモリマップで示されるように、外部コマンドの直後にはBASICのラベル・変数領域がありますが、これらは保存されたままあとのほうへ移動し、新しいコマンドファイルはその前へ挿入されます。

●ファンクション2 [FREEBUFF]

〔機能〕 空いているファイルバッファの番号を得る

〔entry〕 A ← 2

〔exit〕 A → バッファ番号

保存: BC, DE, HL

最大ファイル番号から順に調べられ、空いているバッファ番号を返します。すべてのバッファが使用中のときは、"Bad file number" のエラーを出力します。

●ファンクション 3 [FDDMAIN]

〔機能〕 内部コマンド FDD と同様の処理を行う

〔entry〕 A ← 3

B ← ドライブ番号 1~8 または 0

C ← ディスクモード 1~4 または 0

〔exit〕 保存: BC, DE, H

与えるパラメータと動作は、内部コマンド FDD とまったく同じです。

メモリマップ

RAM の状態は図 5 に示します。

COSMOS 作動中に書き換えられるフックは次の 5 つです。

① ED42, 3, 4 1 文字標準出力の直前

② ED48, 9, A OK 出力の直前

③ ED93, 4, 5 変数アドレス調査の直前

④ EDC9, A, B エラー発生直後

⑤ F302, 3 VRTC 割り込みのエントリアドレス

(ED42) は RST18H からのフックで、標準出力がリダイレクトされたときのみ書き換えられます。

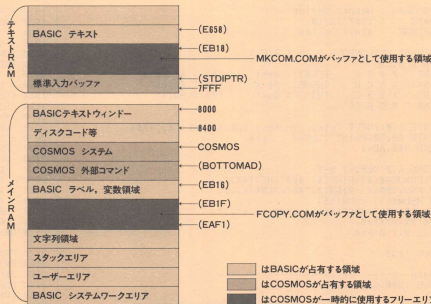
OK 出力の直前には、外部コマンドの離断と標準入出力の強制終了を行います。

変数アドレス調査ルーチン 5ACA からのフック (ED93) では、このルーチンが LET ステートメントから呼び出されたときにだけ (変数名のあとに代入記号「=」がなければ)、COSMOS コマンドが入力されたものとみなして処理を行います。

エラー出力の直前には、外部コマンドの離断を行います。

VRTC 割り込みでは、通常はキースキャンが行われていますが、標準入力リダイレクトされたときのキースキャンを停止して割り込み先を書き換え、ほかのデバイスからのデータをキーバッファへ送り込む処理をさせます。

図 5 メモリマップ



注意事項

●エラーメッセージの説明

1 "Duplicate Definition"

COSMOS がすでに起動されていて、さらに起動しようとしたときにこのエラーが発生します。

2 "Syntax error"

外部コマンドを実行しようとしたとき、コマンドファイルが正しい形式でない場合にこのエラーが出ます。

3 "Bad file number, File already OPEN"

標準入出力、その他のファイルアクセスでは、BASIC システムが用意したファイルバッファを使っています。そのため、BASIC プログラムでファイルを OPEN していないのにバッファが足りなくなることがあり、その際このエラーが発生します。この場合 END を実行したのち、行いたい処理を再実行してみてください。それでもバッファが足りないときは、リセットしてバッファ数を多く設定し直してください。

●COSMOS のコマンドに記述するパラメータのファイル名や数値には、BASIC の変数を使うことができます。

●STOP キーやエラーの発生で処理が停止したときは、標準入出力用以外のファイルバッファはクローズされません。「Break」やエラーメッセージが表示されて止まった状態では、まず END を実行してから再実行してください。こうしてファイルの

クローズを行っておかない場合、出力ファイルが開いたままだとディスクの FAT が未完了のままになっています。そのままディスクを抜いてしまったときは、あとから未完了のファイルを KILL すればディスクは修復されます。

●ディスク以外からの標準入力とすべての標準出力は、ファイルを閉じようとする BASIC ステートメントを実行すると、キーボードとスクリーンにもどつてしまいます。

●標準入力のリダイレクト中は、特別にキーボードからの入力をしたくてデバイスファイル"KYBD:"を OPEN しても、キーボードからの入力はできません。

●BASIC 起動直後でない状態から COSMOS を起動しようとした場合にのみ、暴走することがあるかもしれません。スタックが C900~D32B にあるときに暴走します。BASIC 起動時ならそのようなことはありませんから大丈夫です。また一度 COSMOS を起動してしまえば、この領域にスタックが移動してもかまいません。

●作者の制作環境の都合で、動作未確認の機能があります。RS-232C 通信回線、8 インチフロッピーのシステム、2D タイプの日本語 BASIC での動作は確認されていません。これらを使用するときは、ディスクランチャなどを想定してバックアップをしてからの動作確認をしてから使用してください。

では、COSMOS に関するご意見・ご要望をお待ちしております。

リストA

```

1000 * PROGRAM : Memory Editor
1010 * DATE : 1987/12/10
1020 * AUTHOR : Atus1 Ohshima
1030 *
1040 WIDTH 80,25:CONSOLE 0,25,0,0:CLS
1050 LOCATE 0,20
1060 PRINT " 7 8 9 A [CLR]: Next [A]: Address"
1070 PRINT " 4 5 6 B [HOME]: Back [P]: Print"
1080 PRINT " 1 2 3 C [Q]: Quit [F]: Full"
1090 PRINT " 0 F E D"
1100
1110 LOCATE 0,0:INPUT "clear address":AS=AD=VAL("&h"+AS)
1120 LOCATE 50,20:PRINT "CLEAR":AS
1130 CLEAR 300,AD-1
1140
1150 DEF SNG A,Z:DEFINT B-Y
1160 DEF FNWORDS(X)=RIGHT$("0000"+HEX$(X),4)
1170 DEF FNBYES(X)=RIGHT$("00"+HEX$(X),2)
1180 DIM HSUM(15),VSUM(15)
1190 KEYDAT$="0123456789*+="+CHR$(13)+".,,"
1200 AA=0:XX=0:YY=0:F=0
1210 A=0:X=0:Y=0:F=0
1220
1230 GOSUB 1730
1240
1250 * Main -----
1260 GOSUB 1280:GOTO 1260
1270
1280 LOCATE 5,X*3+F,Y+1:K$=INPUT$(1):N=INSTR(KEYDAT$,K$)
1290 IF N THEN 1440
1300 N=ASC(K$)
1310 IF N=12 THEN 1540
1320 IF N=11 THEN 1580
1330 IF N=28 THEN 1620
1340 IF N=29 THEN 1680
1350 IF N=30 AND Y<0 THEN Y=Y-1:A=A-16:RETURN
1360 IF N=31 AND Y<15 THEN Y=Y+1:A=A+16:RETURN
1370 IF K$="a" THEN 1730
1380 IF K$="f" THEN 1900
1390 IF K$="p" THEN 1820
1400 IF K$="q" THEN 1950
1410 BEEP:RETURN
1420
1430 * Data set -----
1440 N=N-1:PRINT HEX$(N);
1450 D=PEEK(A)
1460 IF F=0 THEN D2=(D AND &HF) OR (N*16) ELSE D2=(D AND &HF0) OR N
1470 POKE A,D2:D3=D2-D
1480 HSUM(Y)=HSUM(Y)+D3:LOCATE 54,Y+1:PRINT FNBYES$(HSUM(Y));
1490 VSUM(X)=VSUM(X)+D3:LOCATE 5+X*3,18:PRINT FNBYES$(VSUM(X));
1500 ZSUM=(ZSUM+D3) MOD 256:LOCATE 54,18:PRINT FNBYES$(ZSUM);
1510 GOTO 1620
1520
1530 * Next -----
1540 AD=AD+256:A=AD:X=0:Y=0:F=0
1550 GOSUB 1990:RETURN
1560
1570 * Back -----
1580 AD=AD-256:A=AD:X=0:Y=0:F=0
1590 GOSUB 1990:RETURN
1600
1610 * Right -----
1620 IF F=0 THEN F=1:RETURN
1630 IF X<15 THEN F=0:X=X+1:A=A+1:RETURN
1640 IF X<15 THEN BEEP 1:F=0:X=0:Y=Y+1:A=A+1:BEEP 0:RETURN
1650 BEEP:RETURN
1660
1670 * Left -----
1680 IF F=1 THEN F=0:RETURN
1690 IF X>0 THEN F=1:X=X-1:A=A-1:RETURN
1700 IF Y>0 THEN F=1:X=15:Y=Y-1:A=A-1:RETURN
1710 BEEP:RETURN
1720
1730 * Input edit address -----
1740 CONSOLE 0,20:CLS
1750 INPUT "edit address":AS:AE=VAL("&h"+AS):AD=INT(AE/256)*256
1760 GOSUB 1990
1770 A=AE:F=0:AS=RIGHT$("0000"+AS$,4)
1780 X=VAL("&h"+MID$(AS$,4,1)):Y=VAL("&h"+MID$(AS$,3,1))
1790 RETURN
1800
1810 * Print -----
1820 CLS
1830 INPUT "start:":AS:AS=VAL("&h"+AS):AS=INT(AS/256)*256
1840 INPUT "end :":AS:AE=VAL("&h"+AS)
1850 GOSUB 2250
1860 AS=AS+256:IF AS<AE THEN 1850
1870 GOSUB 1990:RETURN
1880
1890 * Full -----
1900 D=PEEK(A):AE=INT(A/256)*256+255

```

```

1910 FOR AT=A TO AE:POKE AT,D:NEXT
1920 GOSUB 1990:RETURN
1930
1940 * Quit -----
1950 CONSOLE 0,25:CLS
1960 END
1970
1980 * Display memory sub -----
1990 CLS
2000 PRINT "ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A"
2010 AA=AD
2020 FOR T=0 TO 15:HSUM(T)=0:VSUM(T)=0:NEXT
2030 FOR YY=0 TO 15
2040 LPRINT FNWORDS(AA):CHR$(28);
2050 FOR XX=0 TO 15
2060 D=PEEK(AA):PRINT FNBYES$(D):CHR$(28);
2070 HSUM(YY)=HSUM(YY)+D
2080 VSUM(XX)=VSUM(XX)+D
2090 AA=AA+1
2100 NEXT
2110 PRINT "":FNBYES$(HSUM(YY))
2120 NEXT
2130
2140 PRINT STRING$(57,"-")
2150 PRINT "VSUM ";
2160 ZSUM=0
2170 FOR XX=0 TO 15
2180 PRINT FNBYES$(VSUM(XX)):" ";
2190 ZSUM=(ZSUM+VSUM(XX)) MOD 256
2200 NEXT
2210 PRINT "":FNBYES$(ZSUM)
2220 RETURN
2230
2240 * Printout memory sub -----
2250 AA=AS
2260 LPRINT "ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A"
2270 FOR T=0 TO 15:HSUM(T)=0:VSUM(T)=0:NEXT
2280 FOR YY=0 TO 15
2290 LPRINT FNWORDS(AA):" ";
2300 FOR XX=0 TO 15
2310 D=PEEK(AA):LPRINT FNBYES$(D):" ";
2320 HSUM(YY)=HSUM(YY)+D
2330 VSUM(XX)=VSUM(XX)+D
2340 AA=AA+1
2350 NEXT
2360 LPRINT "":FNBYES$(HSUM(YY))
2370 NEXT
2380
2390 LPRINT STRING$(57,"-")
2400 LPRINT "VSUM ";
2410 ZSUM=0
2420 FOR XX=0 TO 15
2430 LPRINT FNBYES$(VSUM(XX)):" ";
2440 ZSUM=(ZSUM+VSUM(XX)) MOD 256
2450 NEXT
2460 LPRINT "":FNBYES$(ZSUM)
2470 LPRINT:LPRINT:LPRINT
2480 RETURN
2490
2500 * End of Program ...

```

●●●
COSMOS
 ●●●

リスト

COSMOS

1

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C900 18 0F 00 09 AA 08 43 4F 53 4D 4F 53 20 31 2E 30 :25
C910 00 2A 16 EB 2B 7E 2B 6E 67 11 49 8F B7 ED 52 CA :7D
C920 A2 03 21 00 C9 CD 2E D0 ED 53 70 C9 11 00 C9 B7 :64
C930 ED 52 44 4D 2A 16 EB EB CD 0E D0 3E C3 21 A7 C9 :23
C940 32 93 ED 22 94 ED 21 56 CD 32 48 ED 22 49 ED 2A :82
C950 02 F3 22 88 D0 2A D1 EA 23 7E 2B FE 3A 20 0D 7E :03
C960 D6 30 FE 09 30 06 32 43 D1 32 44 D1 CD 7B C9 21 :02
C970 00 00 22 16 EB E1 E1 AF C3 2E 52 E5 21 99 D0 CD :13
C980 50 55 2A 04 D1 46 23 04 20 06 7E 23 66 6F 18 F5 :BA
C990 05 28 0C 7E 23 DF 10 FB 3E 09 DF 23 23 18 E6 21 :4F
C9A0 DB D0 CD 50 55 E1 C9 D5 E5 F5 F3 ED 73 B9 C9 33 :7E
C9B0 33 33 33 33 33 33 33 E1 31 00 00 FB 11 9F 0C B7 :E5
C9C0 ED 52 28 04 F1 E1 D1 C9 F1 E1 D1 22 D4 C9 21 D7 :31
C9D0 C9 E3 E5 21 00 00 C9 7E FE F1 C8 C1 22 0E CA E5 :50
C9E0 2A D4 C9 ED 5B 04 D1 1A 13 3C 20 08 1A 13 47 1A :03
C9F0 58 57 18 F3 3D 28 2D 47 4F E5 1A 13 BE 23 28 02 :FF

```

VSUM 4C 24 CE D4 4C AD 53 07 BD C6 04 B6 D0 A8 B0 E8 :B2

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CA00 0E 00 10 F6 79 B7 28 0D 7E FE 20 28 0D 01 00 00 :4B
CA10 B7 ED 42 28 05 E1 13 13 18 CD E1 1A 13 6F 1A 67 :FD
CA20 E3 2B D7 C9 D1 D5 CD 58 CA 22 47 CA 3A C9 ED 2A :90
CA30 CA ED 32 8D D0 22 8E D0 3E C3 21 43 CD 32 C9 ED :E0
CA40 22 CA ED E1 2B D7 CD 00 00 C5 D5 E5 F5 2A 86 D1 :86
CA50 CD 82 CB F1 E1 D1 C1 C9 ED 53 80 CA C5 D5 11 4E :CA
CA60 D1 3E 22 12 13 3A 43 D1 C6 30 12 13 3E 3A 12 13 :5C
CA70 01 06 00 ED A0 E2 89 CA 7E FE 20 28 0C E5 C5 01 :44
CA80 00 00 B7 ED 42 C1 E1 20 EA D5 21 07 D1 01 04 00 :65
CA90 ED B0 21 0E D1 4E 23 06 00 ED B0 CD 83 CD 32 AF :AF
CAA0 CA 12 13 AF 12 13 12 21 4D D1 D7 CD 98 47 3E 00 :D5
CAB0 32 2D D1 21 01 00 22 31 D1 21 2A D1 D7 CD 98 71 :3F
CAC0 3A AF CA CD 66 CD 11 09 00 19 5E 23 56 23 7E 23 :81
CAD0 66 6F B7 ED 52 22 2F CB 3A AF CA CD 9F CD D1 21 :C5
CAE0 06 D1 4E 23 06 00 ED B0 EB ED 5B 16 EB ED 53 2C :8B
CAF0 CB 73 23 72 23 36 00 AF CD 98 13 CD E4 58 2A 1F :A5

```

VSUM 8D E6 E3 5F E5 9A 55 57 C9 F7 58 7E B2 A0 1E 60 :46

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CB00 EB ED 5B 16 EB B7 ED 52 22 1D CB 2A F1 EA ED 5B :81
CB10 1F EB 1B E5 B7 ED 52 22 26 CB E1 00 00 00 78 :58
CB20 B1 28 02 ED B8 21 00 00 CD 64 CB 21 00 00 11 00 :CF
CB30 00 19 11 1A 00 19 EB 2A 16 EB B7 ED 52 1E 07 D0 :68
CB40 B3 03 21 4D D1 D7 CD BF E5 2A 2C CB CD 2E D0 21 :53
CB50 90 D1 01 1A 00 ED B0 EB CD 82 CB CD 86 5A 2A 2C :21
CB60 CB D1 C1 C9 EB 2A 16 EB 19 22 16 EB 2A 1B EB 19 :C1
CB70 22 1B EB 2A 1D EB 19 22 1D EB 2A 1F EB 19 22 1F :2B
CB80 EB C9 E5 ED 5B 16 EB B7 ED 52 22 9E CB 2A 1F EB :97
CB90 B7 ED 52 44 4D E1 EB 78 B1 28 02 ED B0 21 00 00 :64
CBA0 C3 64 CB CD 7D CC 2B D7 C8 CD D3 11 E5 CD C9 56 :54
CBB0 7E 23 FE 20 D2 A0 4D 4F 06 00 7E 23 66 6F 11 4E :A8
CBC0 D1 3E 22 12 13 ED B0 21 0E D1 4E 23 06 00 ED B0 :07
CBD0 CD 83 CD 32 45 D1 32 47 D1 12 13 AF 12 13 12 21 :DB
CBE0 4D D1 D7 CD 98 47 3A 45 D1 CD 66 CD 23 23 23 7D :7F
CBF0 7E 32 48 D1 FE FF E1 CA 7D CC E5 FE FC 30 68 FE :2F

```

VSUM 37 DA 65 5C 18 1E 21 21 B5 B3 86 21 A9 B1 8F B3 :F5

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CC00 80 D2 A0 4D 3A 45 D1 CD 76 CD EB 21 80 00 B7 ED :CF
CC10 52 EB 1E 07 DA B3 83 45 3E 80 95 67 2E 00 5B :67
CC20 18 EB E5 B7 ED 52 E1 1E 07 DA B3 03 22 49 D1 EB :9B
CC30 0E 01 78 B7 28 27 3A 45 D1 32 2D D1 75 32 31 D1 :8A
CC40 C5 D5 21 2A D1 D7 CD 98 71 3A 45 D1 CD 66 CD 11 :C4
CC50 09 00 19 D1 01 00 01 ED B0 C1 0C 10 D9 3A 45 D1 :98
CC60 CD 9F CD AF 32 45 D1 3E FF 32 CD E6 F3 CD D9 35 :20
CC70 FB AF 32 4C D1 21 F7 CE 22 02 F3 E1 C9 CD 82 CC :BB
CC80 FB C9 F3 3A 47 D1 B7 C8 E5 2A 88 D0 22 02 F3 CD :D3
CC90 D9 35 3A 45 D1 B7 C4 9F CD AF 32 CD E6 32 45 D1 :21
CCA0 32 47 D1 21 00 80 22 49 D1 E1 C9 CD 1E CD 2B D7 :8B
CCB0 C8 CD D3 11 E5 CD C9 56 7E 23 FE 20 D2 00 4D 4F :17
CCC0 06 00 7E 23 66 6F 11 4E D1 3E 22 12 13 ED B0 21 :EF
CCD0 16 D1 4E 23 06 00 ED B0 CD 83 CD 32 46 D1 12 13 :86
CCE0 AF 12 13 12 3A 42 ED 2A 43 ED 32 8A D0 22 8B D0 :B2
CCF0 21 4D D1 D7 CD 98 47 3A 46 D1 CD 66 CD 23 23 23 :7C

```

VSUM 48 0E D5 98 6E CC 1D 6E F6 E4 E0 C2 99 59 33 D2 :FB

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CD00 23 7E FE F8 E1 28 17 E5 FE FC 30 05 FE FA D2 A8 :35
CD10 4D 3E C3 21 CF 32 42 ED 22 43 ED E1 C9 3A :EA
CD20 D1 B7 C8 E5 3A 8A D0 28 8B D0 32 42 ED 22 4A :ED
CD30 3A 46 D1 47 CD 66 CD 7E B7 78 C4 9F CD AF 32 46 :9C
CD40 D1 E1 C9 E5 F5 3A 8D D0 2A 8E D0 32 C9 ED 22 CA :48
CD50 ED F1 E1 C3 49 CA CD 49 CA C5 D5 F5 CD 7D CC CD :E7
CD60 1E CD F1 D1 C1 C9 26 00 6F 29 D5 ED 5B 7F EC 19 :96
CD70 D1 7E 23 66 6F C9 32 27 D1 21 23 D1 CD D3 11 2A :2A
CD80 41 EC C9 C5 D5 E5 3A 7E EC B7 28 0B 47 78 CD 66 :F5
CD90 CD 7E B7 28 05 10 F6 C3 B2 4D 78 E1 D1 C1 C9 E5 :90
CD A0 F5 32 36 D1 21 33 D1 D7 CD 04 4B F1 E1 C9 3A 44 :5F
CD B0 D1 28 11 CD A3 18 3A 7D EC BB 7B 1E 46 DA B3 03 :5F
CD C0 B7 CA B3 03 32 43 D1 C9 01 00 00 28 16 FE 2C 28 :D7
CD D0 0A C5 CD A3 18 C1 47 2B D7 28 08 CF 2C C5 CD A3 :C1
CD E0 18 C1 4F C5 D5 E5 21 62 EF 78 B7 20 11 46 7E B7 :F4
CD F0 28 1C 90 C5 E5 CD 12 CE E1 C1 10 F2 18 10 3D 2B :5F
VSUM FD 06 3E DF C7 73 1E C8 60 27 3B BC 01 45 A3 32 :D9

```

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CE00 2B 96 23 96 23 BE 1E 46 D2 B3 03 CD 12 CE E1 D1 :A6
CE10 C1 C9 47 3E 0B CD C9 37 3E 07 CD D2 37 3E EE CD :FB
CE20 D2 37 AF CD D2 37 3C CD D2 37 CD 47 38 D6 FE 57 :17
CE30 3E 04 28 02 3E 02 B9 DA 06 0B 7A B7 20 74 3E 22 :75
CE40 CD C9 37 CD 47 38 B7 20 0A 3E 21 CD C9 37 3E 01 :65
CE50 CD D2 37 69 26 00 11 3E D1 19 3E 20 CD C9 37 78 :41
CE60 CD D2 37 CD 47 38 32 AC CE 3E 1F CD C9 37 78 CD :3D
CE70 D2 37 7E CD D2 37 3C 3A 14 EF 32 F2 CE 3E 08 32 :40
CE80 14 EF 20 2E 3E 20 CD C9 37 78 CD D2 37 CD 47 38 :16
CE90 FE FF 28 0E 23 23 16 03 0E 02 BE 23 28 14 0C 15 :E0
CE A0 20 F8 3E 1F CD C9 37 78 CD D2 37 3E 01 CD D2 37 :A5
CE B0 18 3F 69 26 00 11 39 D1 19 E5 11 64 EF 2A 60 EF :DC
CE C0 7C 85 80 6F 26 00 19 D1 1A 77 3E 80 04 07 10 FD :67
CE D0 47 3E 18 CD C9 37 CD 47 38 57 79 FE 01 78 28 03 :28
CE E0 B2 18 04 2F E6 0F A2 47 3E 17 CD C9 37 78 CD D2 :14
CE F0 37 3E 04 32 14 EF C9 F3 C5 D5 E5 F5 2A 00 00 11 :19
VSUM 2B 7C F3 91 DB BD B6 CF 25 6B 03 1C 83 9A 8A E5 :83

```

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
CF00 F3 31 B7 ED 52 20 1E DB 09 0F D2 97 CF 07 07 30 :C1
CF10 1B 3A 4C D1 B7 20 25 3A 48 D1 07 38 28 AF CD 76 :1A
CF20 46 78 B9 28 36 F1 E1 D1 C1 C3 87 D0 3A 4C D1 B7 :61
CF30 20 F3 3E 13 32 4C D1 CD F9 30 18 E9 CD F9 30 AF :4F
CF40 32 4C D1 18 E0 3A 45 D1 CD 66 CD 7E B7 28 48 3E :7A
CF50 01 CD F8 45 28 CF CD B1 CF 18 F4 2A 49 D1 CB 7C :E6
CF60 20 35 ED 5B 18 EB B7 ED 52 38 2C 2A 49 D1 3A C2 :3A
CF70 E6 47 F6 02 D3 31 4E 23 79 FE 1A 20 0E 7E B7 20 :AE
CF80 0A 23 CB 7C 28 F7 78 D3 31 18 0C 78 D5 31 22 49 :1A
CF90 D1 79 CD E1 CF 18 8E 3A 45 D1 F5 3A 47 D1 F5 AF :78
CFA0 32 45 D1 CD 82 CC F1 32 47 D1 F1 32 45 D1 C3 25 :BF
CFB0 CF 47 3A 4B D1 4F 78 32 4B D1 FE 0A C2 F9 30 79 :ED
CFC0 FE 0D 78 C2 F9 30 3A 4D D1 B7 78 C2 F9 30 C9 C5 :6E
CFD0 D5 E5 F5 3A 46 D1 CD 66 CD 7E B7 F5 CC 1E CD F1 :D2
CFE0 28 13 2A 88 EC 7C B5 20 0C 3A 46 D1 CD 35 47 F1 :C1
CFF0 F5 DF CD 8B 0F F1 E1 D1 C1 C3 8A D0 FE 01 CA 58 :DD
VSUM 79 77 AD 07 E8 3A 18 5A E5 44 6E C0 06 93 8A 3D :EF

```

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
D000 CA FE 02 CA 83 CD FE 03 CA E3 CD C3 06 0B E5 B7 :CF
D010 ED 52 E1 C8 D5 E5 11 9C 01 B7 ED 52 1E 07 DA B3 :F8
D020 03 E1 D1 D5 ED B0 E1 D1 E5 CD 2E D0 E1 E9 22 39 :AE
D030 D0 23 23 5E 23 56 23 E5 21 00 00 B7 ED 52 22 6C :9A
D040 D0 E1 4E 23 46 2A 39 D0 09 22 8A D0 11 90 D0 1A :A5
D050 13 47 1A 13 32 59 D0 E5 CF 00 E1 23 10 F4 4E 23 :0F
D060 46 23 78 B1 28 1D 5E 23 56 23 E5 21 00 00 EB 19 :DB
D070 22 77 D0 22 7B D0 2A 00 00 19 22 00 00 E1 0B 78 :9F
D080 B1 20 E3 11 00 00 C9 C3 00 00 00 00 00 00 00 :51
D090 06 43 4F 53 4D 4F 53 90 00 00 0A 43 4F 53 4D :FD
D0A0 53 20 76 65 72 73 69 6F 6E 20 31 2E 30 20 5B 44 :E7
D0B0 65 63 2E 31 39 38 37 5D 0D 0A 43 6F 70 79 72 69 :B9
D0C0 67 68 74 20 28 43 29 20 62 79 20 53 4F 46 54 42 :90
D0D0 41 4E 4B 20 4F 68 21 38 38 0D 0A 0D 0A 00 05 44 :B9
D0E0 52 49 56 45 AE CD 03 46 44 44 C8 CD 05 53 54 44 :07
D0F0 49 4E A3 CB 06 53 54 44 4F 55 54 AB CC 03 56 45 :03
VSUM 87 49 15 18 A6 ED 01 2E 77 1B 18 68 2C 3A 34 E8 :53

```


ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
D100	52	7B	C9	00	DE	D0	07	2E	43	4F	4D	22	2C	0C	07	22	:DB
D110	82	85	41	53	23	0C	0C	22	82	4F	55	54	50	55	54	41	:AC
D120	53	23	0C	FF	A5	28	0C	00	00	29	00	23	0C	00	00	2C	:DE
D130	0C	00	00	00	23	0C	00	00	00	03	02	03	09	0A	FF	01	:56
D140	01	11	29	01	01	00	00	00	FF	00	80	00	00	00	00	00	:BC
D150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	AA	D1	:7B
D190	88	D0	AA	D0	49	D1	08	D1	97	D0	00	C9	8E	D1	04	D1	:29
D1A0	43	D1	45	D1	4D	D1	FC	CF	49	8F	43	4F	53	4D	4F	53	:BF
D1B0	BD	00	02	C9	23	C9	26	C9	2A	C9	2D	C9	39	C9	3E	C9	:55
D1C0	47	C9	53	C9	67	C9	6A	C9	6D	C9	7D	C9	83	C9	A0	C9	:C0
D1D0	AD	C9	CC	C9	CF	C9	DD	C9	E1	C9	E5	C9	27	CA	2A	CA	:86
D1E0	33	CA	36	CA	3B	CA	4E	CA	51	CA	5A	CA	5F	CA	66	CA	:B2
D1F0	76	CA	8B	CA	93	CA	9C	CA	9F	CA	A8	CA	B1	CA	B7	CA	:2F

VSUM 59 FB 10 E3 87 A1 7A DF 0C 18 F8 A3 65 79 7C 75 :56

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
D200	BA	CA	C1	CA	C4	CA	D6	CA	D9	CA	DC	CA	E0	CA	EF	CA	:E9
D210	09	CB	18	CB	29	CB	43	CB	4A	CB	4D	CB	50	CB	59	CB	:25
D220	5F	CB	8B	CB	A1	CB	A4	CB	BF	CB	8B	D1	CB	D4	CB	:B3	
D230	D7	CB	E0	CB	E7	CB	EA	CB	F2	CB	F8	CB	05	CC	08	CC	:D9
D240	2D	CC	37	CC	3A	CC	3E	CC	43	CC	4A	CC	4D	CC	5E	CC	:74
D250	61	CC	65	CC	73	CC	76	CC	7E	CC	84	CC	8A	CC	93	CC	:2E
D260	97	CC	9E	CC	A1	CC	A7	CC	AC	CC	C7	CC	D0	CC	D9	CC	:F9
D270	DC	CC	EB	CC	EE	CC	F1	CC	F8	CC	FB	CC	14	CD	1F	CD	:2E
D280	25	CD	28	CD	31	CD	35	CD	3B	CD	3F	CD	46	CD	49	CD	:24
D290	54	CD	57	CD	5D	CD	60	CD	77	CD	7A	CD	8F	CD	A2	CD	:F2
D2A0	A5	CD	AF	CD	C5	CD	F6	CD	0C	CE	57	CE	67	CE	7B	CE	:C0
D2B0	B6	CE	0B	CF	12	CF	18	CF	2A	CF	2D	CF	35	CF	41	CF	:2F
D2C0	46	CF	49	CF	57	CF	5C	CF	6C	CF	8F	CF	93	CF	98	CF	:E0
D2D0	9C	CF	A1	CF	A4	CF	A8	CF	AC	CF	AF	CF	B3	CF	B8	CF	:A7
D2E0	C7	CF	D4	CF	D7	CF	DD	CF	EA	CF	FA	CF	FF	CF	04	D0	:CF
D2F0	09	D0	2A	D0	2F	D0	3F	D0	46	D0	4A	D0	4D	D0	55	D0	:53

VSUM 80 CD 8A CE 17 CE B6 CE 69 CF 38 CF C4 D1 5D D2 :11

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
D300	71	D0	74	D0	97	D0	E4	D0	EA	D0	F2	D0	FB	D0	01	D1	:B9
D310	04	D1	8E	D1	90	D1	92	D1	94	D1	96	D1	98	D1	9A	D1	:98
D320	9C	D1	9E	D1	A0	D1	A2	D1	A4	D1	A6	D1	FF	FF	FF	FF	:A8
D330	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D340	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D350	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D360	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D370	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D380	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D390	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
D3F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0

VSUM 04 65 93 65 BA 65 0B 65 15 65 21 65 85 93 8D 94 :29

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C000 18 12 00 C0 7C 03 4D 4B 43 4F 4D 2E 43 4F 4D 20 :8D
C010 31 2E 30 00 ED 73 20 C2 CD D3 11 22 23 C2 E5 CD :3B
C020 C9 56 06 00 4E 23 7E 23 66 6F 3E 20 B9 DA A0 4D :EA
C030 11 3C C3 ED B0 ED 53 31 C3 3E 02 CD 77 C2 32 15 :6E
C040 C3 21 0E C3 01 0A 00 ED B0 21 3A C3 D7 CD 98 47 :FE
C050 AF 32 28 C3 CD 3A C2 5F CD 3A C2 57 CD 3A C2 6F :4C
C060 CD 3A C2 67 B7 ED 52 22 29 C3 3A 15 C3 32 25 C3 :60
C070 21 22 C3 D7 CD 04 4B 2A 29 C3 ED 5B 18 EB 19 CD :40
C080 91 C2 ED 5B 31 C3 2A 18 EB 22 06 C3 21 02 C3 4E :DB
C090 23 06 00 ED B0 21 3A C3 D7 CD BF EE ED 5B 18 EB :80
C0A0 D5 13 13 CD 26 C2 6F 13 CD 26 C2 67 13 22 2B C3 :71
C0B0 CD 26 C2 6F 13 CD 26 C2 67 D1 19 22 2D C3 CD 91 :AD
C0C0 C2 E5 2A 16 EB 11 EE FF 19 7E 23 66 6F 7E 23 66 :6E
C0D0 6F 46 23 EB E1 1A 13 CD 8F C2 10 F9 22 33 C3 3E :4E
C0E0 02 CD 77 C2 32 15 C3 ED 5B 31 C3 21 09 C3 4E 23 :AC
C0F0 06 00 ED B0 21 3A C3 D7 CD 98 47 AF 32 28 C3 06 :16

```

VSUM 12 7A 27 68 F2 A8 1D 39 CE 9F 9E 30 2F AF 66 EF :79

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C100 06 CD 3A C2 18 FB 21 01 00 22 35 C3 ED 5B 18 EB :61
C110 2A 2B C3 B7 ED 52 22 37 C3 2A 33 C3 23 23 13 13 :B6
C120 E5 2A 37 C3 19 44 4D E1 79 CD 8F C2 78 CD 8F C2 :C1
C130 E5 13 13 D5 CD 3A C2 6F CD 3A C2 67 ED 5B 2B C3 :7E
C140 B7 ED 52 CA EB C1 22 39 C3 D1 E1 CD 26 C2 4F CD :0D
C150 3A C2 47 B9 28 32 D5 E5 2A 37 C3 19 EB E1 7B CD :61
C160 8F C2 7A CD 8F C2 D1 13 E5 68 CD 26 C2 47 CD 3A :1D
C170 C2 67 D5 B7 ED 42 ED 4B 39 C3 B7 ED 42 20 6C 2A :B4
C180 35 C3 23 22 35 C3 D1 E1 13 E5 2A 2D C3 B7 ED 52 :EF
C190 E1 38 58 20 B6 22 2F C3 2A 33 C3 ED 5B 35 C3 7B :36
C1A0 CD 8F C2 7A CD 8F C2 3A 15 C3 32 25 C3 21 22 C3 :E8
C1B0 D7 CD 04 4B 2A 18 EB 22 1C C3 EB 2A 2F C3 B7 ED :CC
C1C0 52 22 20 C3 2A 16 EB 11 EC FF 19 7E 23 66 6F ED :FA
C1D0 5B 31 C3 13 01 03 00 ED B0 21 18 C3 4E 23 06 00 :76
C1E0 ED B0 21 3A C3 D7 CD C2 EE E1 C9 2A 16 EB 11 E6 :DB
C1F0 FF 19 7E 23 66 6F 7E 23 66 6F 11 08 E8 B7 ED 52 :FB

```

VSUM 8F 80 F2 52 A8 AD EA E7 72 94 F6 84 09 AB E4 23 :B4

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C200 21 BF C2 28 0A 3A 70 85 FE 03 38 03 21 D3 C2 CD :C2
C210 50 55 3A 15 C3 32 25 C3 21 22 C3 D7 CD 04 4B 31 :FB
C220 00 00 21 00 00 C9 F3 3A C2 6E 32 34 C2 F6 02 D3 :B2
C230 31 1A F5 3E 00 D3 31 F1 FB C9 C5 05 E5 3A 28 C3 :DB
C240 B7 20 0D 3A 15 C3 32 25 C3 21 22 C3 D7 CD 98 71 :C3
C250 CD 67 C2 11 09 00 19 3A C2 5F 16 00 19 3C 32 4A :4A
C260 28 C3 7E E1 D1 C1 C9 2A 7F EC 3A 15 C3 87 5F 16 :48
C270 00 19 7E 23 66 6F C9 F5 22 8A C2 2A 16 EB 2B 2B :3C
C280 2B 7E 2B 6E 67 F1 22 8D C2 21 00 00 C3 00 00 77 :66
C290 23 F5 D5 E5 2A 16 EB 11 EA FF 19 7E 23 66 6F 5E :E4
C2A0 23 56 E1 E5 B7 ED 52 30 04 E1 D1 F1 C9 3A 15 C3 :E7
C2B0 32 25 C3 21 22 C3 D7 CD 04 4B 1E 07 C3 B3 03 0D :BE
C2C0 0A 42 61 64 20 66 69 6C 65 20 66 6F 72 6D 61 74 :7A
C2D0 0D 0A 00 0D 0A 83 74 83 40 83 43 83 8B 82 CC 83 :8D
C2E0 74 83 48 81 7C 83 7D 83 62 83 67 82 AA 90 B3 82 :FC
C2F0 B5 82 AD 82 A0 82 E8 82 DC 82 B9 82 F1 81 42 0D :4C

```

VSUM 31 D0 D7 97 D2 A0 0E 80 FF 22 40 67 4F B2 3E A3 :19

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C300 0A 00 06 22 2C 0C 00 00 00 00 2E 4F 46 53 22 82 :32
C310 85 41 53 23 8C 00 00 00 0A 22 2C 0C 00 00 2C 0C :E4
C320 00 00 00 23 8C 00 00 00 00 00 00 00 00 00 00 :2F
C330 00 00 00 00 00 00 00 00 00 00 00 22 00 00 00 :22
C340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C370 00 00 00 00 00 00 00 00 00 00 00 00 43 4F 53 4D :32
C380 4F 53 60 00 02 C0 16 C0 1C C0 31 C0 37 C0 3C C0 :5A
C390 3F C0 42 C0 4A C0 52 C0 55 C0 59 C0 5D C0 61 C0 :89
C3A0 68 C0 6B C0 6E C0 71 C0 78 C0 80 C0 84 C0 8A C0 :B8
C3B0 8D C0 96 C0 A4 C0 A9 C0 AE C0 B1 C0 B6 C0 BC C0 :41
C3C0 BF C0 D8 C0 DD C0 E2 C0 E5 C0 E9 C0 EC C0 F5 C0 :05
C3D0 FD C0 02 C1 0A C1 11 C1 17 C1 1A C1 22 C1 2A C1 :9E
C3E0 2E C1 35 C1 39 C1 3E C1 44 C1 47 C1 4C C1 50 C1 :09
C3F0 59 C1 60 C1 64 C1 6B C1 6F C1 78 C1 80 C1 84 C1 :7B

```

VSUM 55 D6 6B AB 26 0F 1E 03 50 33 D7 80 31 A5 77 DE :9C

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C400 8B C1 96 C1 99 C1 9D C1 A1 C1 A5 C1 A8 C1 AB C1 :F8
C410 AE C1 B8 C1 BC C1 C2 C1 D1 C1 DA C1 E3 C1 01 C2 :7C
C420 0D C2 13 C2 16 C2 19 C2 2B C2 3E C2 44 C2 47 C2 :53
C430 4A C2 51 C2 58 C2 60 C2 6B C2 79 C2 87 C2 AE C2 :7C
C440 B1 C2 B4 C2 FF FF FF FF FF FF FF FF FF FF FF FF :DD
C450 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C460 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C470 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C480 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C490 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
C4F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
-----
VSUM 36 BD 5B BD B7 FA CC FA FC FA 2A FA 4A FA 95 FB :70

```

リスト 3 FCOPY.COM

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C000 18 12 00 C0 45 04 46 43 4F 50 59 2E 43 4F 4D 20 :E1
C010 31 2E 30 00 CD F2 C1 3E 02 CD 01 C2 32 7C C3 32 :82
C020 C2 C3 CD D3 11 E5 CD C9 56 11 44 C3 CD 0C C2 21 :DB
C030 74 C3 CD 23 C2 E1 CF 2C CD D3 11 E5 CD C9 56 11 :58
C040 80 C3 CD 0C C2 D5 21 42 C3 D7 CD 98 47 3E 02 CD :69
C050 01 C2 32 BD C3 32 C7 C3 D1 21 B0 C3 CD 23 C2 21 :69
C060 7E C3 D7 CD 98 47 3A C2 C3 CD 41 C2 23 23 23 23 :DF
C070 7E 32 1C C4 23 23 23 7E E6 FD 32 1E C4 3A C7 C3 :32
C080 CD 41 C2 23 7E 32 1F C4 23 23 23 7E 32 1D C4 06 :86
C090 00 3A 1C C4 07 38 07 3A 1D C4 07 38 01 05 78 32 :6A
C0A0 20 C4 3A 1C C4 07 DA 80 C1 CD 9A 55 2A F1 EA ED :CE
C0B0 5B 1F EB B7 ED 52 7C B7 CA 52 C2 32 21 C4 3A C2 :7F
C0C0 C3 CD 2E C2 22 26 C4 2A 26 C4 CB 7C 20 2A 3A 21 :8C
C0D0 C4 4F 06 00 B7 ED 42 22 26 C4 30 06 09 4D 79 B7 :C7
C0E0 28 16 41 ED 5B 1F EB CD 5A C2 10 FB 41 ED 5B 1F :6D
C0F0 EB CD 7C C2 10 FB 18 CF CD A1 C2 3A 20 C4 B7 E1 :CE
-----
VSUM DE 9D B0 3B 9F 1D 6D D8 EF B4 F2 C7 12 5D FB 17 :44

```

```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
C100 C8 E5 3A 1D C4 3C 32 1D C4 3E 05 CD B8 C2 32 23 :F6
C110 C4 57 3E 02 CD B8 C2 32 24 C4 5F 06 00 7A FE 12 :AB
C120 20 05 7B 3D 20 01 04 78 32 22 C4 06 00 7A FE 12 :22
C130 20 05 7B B7 20 01 05 78 32 25 C4 0E 00 0C 3E 0A :72
C140 CD B8 C2 B9 20 05 1E 35 C3 B3 03 CD D1 C2 AF CD :CD
C150 41 C2 11 09 00 19 06 10 7E B7 28 0F 3C 28 E7 EB :EE
C160 21 0A 00 19 3A 1F C4 BE EB 28 08 11 10 00 19 10 :84
C170 E7 18 CA 11 09 00 19 3A 1E C4 77 CD 0B C3 E1 C9 :D4
C180 3A C2 C3 CD 95 C1 38 08 3A C7 C3 CD D5 C1 18 F0 :51
C190 CD A1 C2 E1 C9 4F C5 3A 1C C4 3C C1 79 28 0F CD :82
C1A0 35 47 11 00 00 CD D7 4C CD 8B 0F C3 C9 56 CD C8 :5B
C1B0 5F 38 13 21 B9 E9 06 FF 7E 23 04 B7 20 FA 78 32 :92
C1C0 28 C4 21 28 C4 C9 CD CE 35 28 FB FE 0D 20 F7 AF :86
C1D0 32 CA E6 37 C9 E5 CD 35 47 E1 46 23 78 B7 28 09 :BA
C1E0 7E 23 66 6F 7E 23 DF 10 FB 3E 0D DF 3E 0A DF C3 :15
C1F0 8D 0F E5 2A 16 EB 2B 11 44 C4 01 1A 00 ED B8 E1 :8F
-----
VSUM E0 84 06 C6 6C B5 7C 2D F2 E3 F7 C3 DA 76 1E F5 :EC

```


ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
C200	C9	E5	2A	41	C4	22	0A	C2	E1	C3	00	00	C5	E5	F5	06	:14
C210	00	4E	23	7E	23	66	6F	3E	20	B9	DA	A0	4D	ED	B0	F1	:53
C220	E1	C1	C9	C5	E5	4E	23	06	00	ED	B0	E1	C1	C9	C5	D5	:2E
C230	F5	32	18	C4	21	14	C4	CD	D3	11	2A	41	EC	F1	D1	C1	:87
C240	C9	D5	F5	2A	7F	EC	87	5F	16	00	19	7E	23	66	6F	F1	:A4
C250	D1	C9	CD	A1	C2	1E	07	C3	B3	03	C5	E5	F5	D5	CD	86	:2F
C260	5A	21	BF	C3	D7	CD	98	71	3A	C2	C3	CD	41	C2	11	09	:53
C270	00	19	D1	01	00	01	ED	B0	F1	E1	C1	C9	C5	E5	F5	D5	:59
C280	CD	86	5A	3A	C7	C3	CD	41	C2	11	09	00	19	D1	EB	01	:31
C290	00	01	ED	B0	E5	21	C4	C3	D7	CD	A6	71	D1	F1	E1	C1	:4A
C2A0	C9	C5	D5	E5	F5	21	BF	C3	D7	CD	04	4B	21	C4	C3	D7	:52
C2B0	CD	04	4B	F1	E1	D1	C1	C9	C5	D5	E5	32	10	C4	3A	1D	:25
C2C0	C4	32	0C	C4	21	08	C4	CD	D3	11	3A	41	EC	E1	D1	C1	:3E
C2D0	C9	C5	D5	E5	F5	3A	1D	C4	32	CD	C3	32	E0	C3	3A	22	:4B
C2E0	C4	32	D1	C3	3A	23	C4	32	D5	C3	32	E4	C3	79	32	D9	:02
C2F0	C3	32	E8	C3	3A	25	C4	B7	21	CA	C3	28	03	21	DD	C3	:14

VSUM	0A	A9	81	C6	11	22	ED	20	F8	0B	A0	28	8A	F6	60	17	:FC
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
C300	CD	D3	11	CD	C9	56	F1	E1	D1	C1	C9	C5	D5	E5	F5	3A	:78
C310	1D	C4	32	ED	C3	32	FD	C3	3A	22	C4	32	F1	C3	3A	23	:18
C320	C4	32	F5	C3	32	01	C4	79	32	F9	C3	32	05	C4	3A	25	:66
C330	C4	B7	21	EB	C3	28	03	21	FB	C3	D7	CD	98	EE	F1	E1	:50
C340	D1	C1	C9	22	00	00	00	00	00	00	00	00	00	00	00	00	:7D
C350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C370	00	00	00	00	0A	22	82	85	41	53	23	0C	00	00	00	22	:18
C380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C3A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C3B0	0F	22	82	4F	55	54	50	55	54	41	53	23	0C	00	00	00	:67
C3C0	23	0C	00	00	00	23	0C	00	00	00	EC	28	0C	00	00	2C	:AA
C3D0	0C	00	00	2C	0C	00	00	2C	0C	00	00	29	00	EC	28	0C	:C5
C3E0	00	00	2C	0C	00	00	2C	0C	00	00	29	00	0C	00	00	2C	:D1
C3F0	0C	00	00	2C	0C	00	00	2C	0C	00	00	00	0C	00	00	2C	:B4

VSUM	8D	6F	D0	3D	F8	4A	BF	7C	E5	33	B2	76	93	46	82	15	:36
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
C400	0C	00	00	2C	0C	00	00	00	FF	D0	28	0C	00	00	2C	0C	:7F
C410	00	00	29	00	FF	A5	28	0C	00	00	29	00	00	00	00	00	:2A
C420	00	00	00	00	00	00	00	00	00	B9	E9	00	00	00	00	00	:A2
C430	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C440	00	00	00	00	00	43	4F	53	4D	4F	53	6D	00	02	C0	15	:18
C450	C0	1A	C0	1D	C0	20	C0	2A	C0	2D	C0	30	C0	33	C0	40	:51
C460	C0	43	C0	47	C0	50	C0	53	C0	56	C0	5A	C0	5D	C0	60	:9A
C470	C0	67	C0	6A	C0	72	C0	7B	C0	7E	C0	81	C0	86	C0	8D	:D0
C480	C0	92	C0	98	C0	A0	C0	A3	C0	A7	C0	B9	C0	BC	C0	BF	:48
C490	C0	C2	C0	C5	C0	C8	C0	CF	C0	D8	C0	E8	C0	F2	C0	F9	:C9
C4A0	C0	FC	C0	03	C1	07	C1	0C	C1	0F	C1	15	C1	18	C1	19	:7D
C4B0	C1	39	C1	41	C1	4C	C1	50	C1	65	C1	78	C1	7C	C1	81	:F8
C4C0	C1	84	C1	89	C1	8C	C1	91	C1	98	C1	C0	C1	C3	C1	F8	:45
C4D0	C1	03	C2	06	C2	32	C2	35	C2	53	C2	62	C2	69	C2	6C	:09
C4E0	C2	84	C2	87	C2	96	C2	A6	C2	AD	C2	BC	C2	BF	C2	C2	:41
C4F0	C2	C5	C2	D6	C2	D9	C2	DC	C2	DF	C2	E2	C2	E5	C2	E8	:EE

VSUM	53	1D	71	87	54	B2	C0	6D	95	43	D6	72	49	2A	35	BE	:21
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

ADDR	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
C500	C2	EB	C2	EF	C2	F2	C2	F5	C2	F9	C2	FE	C2	10	C3	13	:EC
C510	C3	16	C3	19	C3	1C	C3	1F	C3	22	C3	25	C3	29	C3	2C	:1E
C520	C3	2F	C3	33	C3	38	C3	3F	FF	FF	FF	FF	FF	FF	FF	FF	:9D
C530	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C540	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C550	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C560	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C570	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C580	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C590	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C5F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0

VSUM	3B	23	3B	2E	3B	39	3B	06	77	0D	77	15	77	2B	78	31	:D7
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

通信データコンバータ

NICE ACCESS

ちょっと便利なプログラムはいくらあってもいいもの。このプログラムは、パソコン通信の際のアップロード、ダウンロードに役立つファイルコンバータです。すべてBASICで書かれており、200ラインとすこしのShortプログラムです。すぐ打ち込めて役に立つ、こんなありがたいものはありません。!

古村勇一

プログラムの機能と概要

88でパソコン通信を楽しんでいる方も多いと思いますが、今回はアップロードやダウンロードに役立つファイルコンバータを作ってみました。

①ダウンロードの際に記録された改行コードや復帰コードを除去する(メニュー1)

②アップロードする文章の1行の文字数をそろえる(メニュー2)

③文章中の英数カナ文字を半角文字/全角

文字にそろえる(メニュー3)

必要なハードウェア

① PC-8800シリーズのパソコン本体

② フロッピードライブ (最低1台必要)

③ ディスプレイ (低解像度可)

プログラムの入力と実行方法

① このプログラムはV1/V2/V3のいずれのBASICモードでも実行できます

② "How many files?"への設定は、"2"

③ を入力してください

③ リスト1のプログラムを入力します。行末でRETURNキーを押すのを忘れないように

④ プログラムを入力したら、

save "FLCONV" ②

て、プログラムをセーブします

⑤ RUN ②

で実行です。使い方は、表示されるメニューに従ってください

注意点

① "CR"は改行コード、"LF"は復帰コードです

② メニュー2で1行の文字数を指定しますが、これは半角文字数で計算してください

③ ファイル内の漢字はシフト JIS 漢字コードで扱っています。ダウンロードする際に通信ソフトで「シフト JIS 漢字コード」で格納するように指定してください

④ ファイルの格納先を"SCRN:"や"LPT:"に指定すると、ディスプレイに表示したり、プリンタに出力することができす

リスト1

```

1000 '
1010 ' File Converter ver 1.0 by Megumu Taketatsu
1020 '
1030 ON ERROR GOTO 2948
1040 PRINT
1050 PRINT "File Converter ver.1.0"
1060 '
1070 ' < Main >
1080 '
1090 PRINT
1100 PRINT " << Main Menu >>"
1110 '
1120 PRINT " 1 --- CR,LF 変換"
1130 PRINT " 2 --- LF 変換"
1140 PRINT " 3 --- 全角/半角変換"
1150 PRINT " 4 --- 終了"
1160 PRINT
1170 LINE INPUT "2:1" : "1:WORKS"
1180 ON VAL(WORKS) GOTO 1230,1720,2420,2900
1190 PRINT "2:1" 13"
1200 GOTO 1170
1210 '
1220 '
1230 ' < 1 --- CR,LF 変換 >
1240 '
1250 PRINT
1260 PRINT " < CR,LF 変換 >"
1270 '
1280 PRINT " 1 --- CR 変換"
1290 PRINT " 2 --- LF 変換"
1300 PRINT " 3 --- CR,LF 変換"
1310 PRINT " 4 --- Main Menu"
1320 PRINT
1330 LINE INPUT "2:1" : "1:WORKS"
1340 ON VAL(WORKS) GOTO 1380,1480,1420,1070
1350 PRINT "2:1" 13"
1360 GOTO 1230
1370 '
1380 '
1390 '
1400 '
1410 '
1420 '
1430 '
1440 LINE INPUT "2:1" 2333 746 (RETURN+FILES : **276) : "INPUT FILES"
1450 IF INPUT FILES="" THEN 1090
1460 IF INPUT FILES="" OR INPUT FILES="1" THEN FILES 1:GOTO 1430
1470 IF INPUT FILES="2" THEN FILES 2:GOTO 1430
1480 OPEN INPUT FILES FOR INPUT AS #1
1490 '
1500 LINE INPUT "2:1" 2333 746 (RETURN+FILES : **276) : "OUTPUT FILES"
1510 IF OUTPUT FILES="" THEN 1090
1520 IF OUTPUT FILES="" OR OUTPUT FILES="1" THEN FILES 1:GOTO 1500
1530 IF OUTPUT FILES="2" THEN FILES 2:GOTO 1500
1540 IF OUTPUT FILES="SCRN:" OR OUTPUT FILES="LPT:" THEN MODE=0: ELSE MODE=0+0
1550 OPEN OUTPUT FILES FOR OUTPUT AS #2
1560 '
1570 PRINT
1580 PRINT "2:1" 2333 746 (RETURN+FILES : **276)
1590 IF MODE=0 THEN PRINT
1600 COUNT=1:0:COUNT=2:0
1610 WHILE NOT (EOF(1))
1620 WORKS=INPUT$(1)

```

```

1638 IF NOT (MODE=1) AND WORKS<CHR$(13)) OR (MODE=2) AND WORKS<CHR$(10)) THEN PRINT #2,WORKS: ELSE COUNT=2*COUNT+1
1640 IF MODE=0 THEN COUNT=1*COUNT+1:LOCATE 14:PRINT COUNT:1:COUNT,2:
1650 MENU
1660 PRINT "... 000 0000000"
1670 CLOSE #2
1680 CLOSE #1
1690 GOTO 1290
1700 '
1710 '
1720 ' < 2 --- CR,LF 7 16 >
1730 '
1740 PRINT
1750 PRINT " < CR,LF 7 999 >"
1760 PRINT
1770 PRINT " 1 --- CR 7 999"
1780 PRINT " 2 --- LF 7 999"
1790 PRINT " 3 --- CR,LF 7 999"
1800 PRINT " 4 --- Main Menu ^"
1810 PRINT
1820 LINE INPUT "???" > "WORKS"
1830 ON VAL(WORKS) GOTO 1870,1890,1910,1870
1840 PRINT "???" > " "
1850 GOTO 1820
1860 '
1870 MODE=1:MODE=2=0:GOTO 1920
1880 '
1890 MODE=1=0:MODE=2=1:GOTO 1920
1900 '
1910 MODE=1=1:MODE=2=1
1920 '
1930 LINE INPUT "!!" > " " (==>000) > "WORKS"
1940 IF WORKS="" THEN 2480
1950 L=LEN(VAL(WORKS))
1960 IF L=LEN=0 THEN PRINT "!!" > " " GOTO 1930
1970 '
1980 LINE INPUT "???" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
1990 IF INPUT,FILES="" THEN 2480
2000 IF INPUT,FILES="" OR INPUT,FILES="1" THEN FILES 1:GOTO 1970
2010 IF INPUT,FILES="2" THEN FILES 2:GOTO 1970
2020 OPEN INPUT,FILES FOR INPUT AS #1
2030 '
2040 LINE INPUT "???" > " " (RETURN=FILES: ==>000) > "OUTPUT,FILES"
2050 IF OUTPUT,FILES="" THEN 2390
2060 IF OUTPUT,FILES="" OR OUTPUT,FILES="1" THEN FILES 1:GOTO 2030
2070 IF OUTPUT,FILES="2" THEN FILES 2:GOTO 2030
2080 IF OUTPUT,FILES="SCRN" OR OUTPUT,FILES="scrn" THEN MODE=0: ELSE MODE=0=0
2090 OPEN OUTPUT,FILES FOR OUTPUT AS #2
2100 '
2110 PRINT
2120 PRINT "... 000 29+ ...!"
2130 IF MODE=0:1 THEN PRINT
2140 COUNT=1=0:COUNT=2=L.DATAS=""
2150 WHILE NOT (EOF(1))
2160 WORKS=INPUT$(1)
2170 IF WORKS<CHR$(13) AND WORKS<CHR$(10) THEN L.DATAS=L.DATAS+WORKS
2180 IF L=LEN(L.DATAS) THEN 2270
2190 LAST,CHR(ASC(RIGHT$(L.DATAS,1)))
2200 IF NOT ((LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80)))) THEN 2230
2210 IF NOT ((LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80)))) THEN 2230
2220 WORKS=INPUT$(1)
2230 IF WORKS<CHR$(13) AND WORKS<CHR$(10) THEN L.DATAS=L.DATAS+WORKS
2240 '
2250 PRINT "???" > " "
2260 IF MODE=1:1 THEN PRINT #2,CHR$(13):
2270 IF MODE=2:1 THEN PRINT #2,CHR$(10):
2280 L.DATAS=""COUNT=2=COUNT+1
2290 '
2290 IF MODE=0 THEN COUNT,1=COUNT,1+1:LOCATE 14:PRINT COUNT:1:COUNT,2:
2300 WEND
2310 IF L=LEN(L.DATAS) THEN 2360
2320 PRINT #2,L.DATAS:
2330 IF MODE=1:1 THEN PRINT #2,CHR$(13):
2340 IF MODE=2:1 THEN PRINT #2,CHR$(10):
2350 COUNT=2=COUNT+1
2360 IF MODE=0 THEN COUNT,1=COUNT,1+1:LOCATE 14:PRINT COUNT:1:COUNT,2:
2370 PRINT "... 000 0000000"
2380 CLOSE #2
2390 CLOSE #1
2400 GOTO 1720
2410 '
2420 ' < 3 --- 000/000000 000 >
2430 '
2440 PRINT
2450 PRINT " < 000/000000 000 >"
2460 PRINT
2470 PRINT " 1 --- 000/000000 000"
2480 PRINT " 2 --- 000/000000 000"
2490 PRINT " 3 --- Main Menu ^"
2500 PRINT
2510 LINE INPUT "???" > "WORKS"
2520 ON VAL(WORKS) GOTO 2560,2580,1870
2530 PRINT "???" > " "
2540 GOTO 2510
2550 '
2560 MODE=1=0:GOTO 2590
2570 '
2580 MODE=1=1
2590 '
2600 LINE INPUT "???" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
2610 IF INPUT,FILES="" THEN 2800
2620 IF INPUT,FILES="" OR INPUT,FILES="1" THEN FILES 1:GOTO 2590
2630 IF INPUT,FILES="2" THEN FILES 2:GOTO 2590
2640 OPEN INPUT,FILES FOR INPUT AS #1
2650 '
2660 LINE INPUT "???" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
2670 IF INPUT,FILES="" THEN 2870
2680 IF INPUT,FILES="" OR INPUT,FILES="1" THEN FILES 1:GOTO 2650
2690 IF OUTPUT,FILES="2" THEN FILES 2:GOTO 2650
2700 IF OUTPUT,FILES="SCRN" OR OUTPUT,FILES="scrn" THEN MODE=0: ELSE MODE=0=0
2710 OPEN OUTPUT,FILES FOR OUTPUT AS #2
2720 '
2730 PRINT
2740 PRINT "... 000 29+ ...!"
2750 IF MODE=0:1 THEN PRINT
2760 COUNT=1=0
2770 WHILE NOT (EOF(1))
2780 WORKS=INPUT$(1)
2790 IF ((LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80))) OR (LAST,CHR(ASC(80)))) THEN WORKS=WORKS+INPUT$(1):K,FLAG=1 ELSE K
2800 IF MODE=1=0 THEN IF K,FLAG=0 THEN PRINT #2,KACN=WORKS: ELSE PRINT #2,WORKS:
2810 IF MODE=1=1 THEN IF K,FLAG=1 THEN PRINT #2,KACN=WORKS: ELSE PRINT #2,WORKS:
2820 IF MODE=0 THEN COUNT,1=COUNT,1+1:LOCATE 14:PRINT COUNT,1:
2830 WEND
2840 '
2850 PRINT "... 000 0000000"
2860 CLOSE #2
2870 CLOSE #1
2880 GOTO 2420
2890 '
2900 ' < 4 --- 000 >
2910 PRINT
2920 PRINT
2930 '
2940 ' < 5 --- 000 >
2950 '
2960 IF ERL=1480 THEN PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
2970 IF ERL=1550 THEN PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"

```

```

2980 IF ERL=1670 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
2990 IF ERL=2020 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3000 IF ERL=2020 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3010 IF ERL=2060 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3020 IF ERL=2380 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3030 IF ERL=2640 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3040 IF ERL=2640 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3050 IF ERL=2710 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3060 IF ERL=2860 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3070 IF ERL=2900 THEN PRINT:PRINT "!!" > " " (RETURN=FILES: ==>000) > "INPUT,FILES"
3080 RESUME NEXT

```


あなたの部屋にも環境音楽を

BACH 管弦楽デモンストレーション

BGM を環境として楽しもうというのが、最近流行の「環境音楽」です。パソコンをプログラムやゲーム以外にもっと活用してみませんか。部屋の中で放っておくだけでクラシックを奏でるパソコンと

いうのも、88ユーザーだけが味わえる楽しみ方ではないでしょうか。このプログラムは、バッハの「管弦楽組曲第3番」の全楽章をBASICで組んだもので、今回と次回2度に分けて掲載します。

小林大二

● 環境音楽プログラム論

みなさんは「環境音楽」という言葉をご存じですか？ インテリアが部屋の環境として存在するように、BGMを環境として楽しもうというのが環境音楽。といっても、普通の音楽と何が違うのかと言われると返答に困るのですが。最近ではビデオの普及で、音楽と映像と一緒に楽しむ「環境ビデオ」なるもの（浜辺や風景をええんと映し続けるだけといったもの。まあ、テレビの台情報をつなぎに流れる映像のようなものと思えば、間違いありません）も発売されています。

そこで環境音楽プログラム論。パソコンをプログラムを組んだりゲームで遊ぶ以外には、ただの箱として置いておくのもったいないもの。とくにPC-8800シリーズのSR以降の機種にはFM音源が搭載されていますので、これを使わないではありませぬ。今回は、クラシック、それもバッハを環境音楽プログラムにしてみました。

このプログラムは学校の文化祭への出演を目的に、バッハの「管弦楽組曲第3番」全楽章を環境音楽プログラムにしたのですが、原曲があまりに長いために1つのプログラムにすることはできませんでした。そこで音楽データを各楽章ごとに分けてディスクに格納し、演奏時に必要な部分だけ読み出す方法を取っています。

今回は音楽の全データを紹介し、次回に

すべての音楽データを操作し、各曲についての説明文が表示されるプログラムと、説明文のデータプログラムを紹介します。といっても次回までデータを打ち込むだけというのも残念ですので、演奏用の簡単なプログラムも作ってみました。

● 必要なハードウェア

- ① PC-8800シリーズでV2モードのBASICフロッピードライブが最低1台は必要。
- ② フォーマットしたディスク3枚
1枚はプログラム保存用、もう1枚は音楽データの保存用に、もう1枚をシステムに使います。
- ③ できればアンプとスピーカ
内蔵スピーカだけでも演奏できないことはありませんが、アンプとスピーカに接続すると、より厚みのある音を楽しむことができます。

● リストの打ち込み方と音楽データの作成

- ① BASICモード切り替えスイッチを「V2」にして電源を入れます。V2モードのないPC-8800シリーズでは実行できません。
- ② “How many files?”への設定は、6 ⑥
を入力してください。
- ③ フォーマットして、システムの入ったフ

ロッピーを2枚用意します。

- ④ 音楽データ作成プログラム（リスト1～12）を入力します。リストごとの行末で⑥を押すのを忘れないように。
- ⑤ 1つのリストを入力後、プログラム保存用のフロッピーをドライブ1に挿入し、SAVE”（ファイル名）⑥を入力します。ファイル名は各プログラムの30行にある名前を指定してください。
- ⑥ プログラムをディスクにセーブしたら、今度は音楽データ保存用のフロッピーをドライブ1に挿入し、RUN ⑥を入力します。“OK”が表示されたら、音楽データ作成完了です。
- ⑦ 音楽データ作成プログラム（リスト1～12）を入力するたびに、⑤～⑥の手順を繰り返します。プログラム保存用のフロッピーと音楽データ保存用のフロッピーを間違えないようにしてください。

● 演奏方法

- ① 演奏用のプログラム（リスト13～14）を入力します。
- ② プログラムを入力したら、データ保存用のフロッピーにセーブしてください。
- ③ 第1楽章を演奏するには、リスト13のプログラムを、それ以外の楽章を演奏するにはリスト14のプログラムを使います。
- ④ 音楽データ保存用のフロッピーをドライブ1に挿入します。

⑤第1章の演奏は、

LOAD "list p 1" ⊖

でプログラムをロードします。ドライブ1に音楽データ保存用のフロッピーを挿入したら、

RUN ⊖

で演奏が始まります。

⑥第1章以外の演奏は、音楽データ保存用

LOAD "list p 2" ⊖

でプログラムをロードします。ドライブ1に音楽データ保存用のフロッピーを挿入したら、

RUN ⊖

を入力します。画面に、

Select Music Number (1-6)

1...DATA. M・2

2...DATA. M・3

3...DATA. M・4

4...DATA. M・5

5...DATA. End

6...DATA. Ope

と表示されたら、演奏する曲の番号を選択します。

「DATA. M・2」は第2章楽、「DATA. M・4」は第4章楽、「DATA. M・5」は第5章楽、

「DATA. End」はエンディング曲、「DATA. Ope」はオープニング曲です。

リストの説明

各リストの内容はつぎのとおりです。

①リスト1(プログラム名:MUSIC 1.M8

A)

第1章楽第1バイオリンのパート。音楽データ"DATA A. M・1"を作成。

②リスト2(プログラム名:MUSIC 1.M8

B)

第1章楽第2バイオリンのパート。音楽データ"DATA B. M・1"を作成。

③リスト3(プログラム名:MUSIC 1.M8

C)

第1章楽ピアノのパート。音楽データ"DATA C. M・1"を作成。

④リスト4(プログラム名:MUSIC 1.M8

D)

第1章楽ハーブシコードのパート。音楽データ"DATA D. M・1"を作成。

⑤リスト5(プログラム名:MUSIC 1.M8

E)

第1章楽ホルン1のパート。音楽データ"DATA E. M・1"を作成。

⑥リスト6(プログラム名:MUSIC 1.M8

F)

第1章楽ホルン2のパート。音楽データ"DATA F. M・1"を作成。

⑦リスト7(プログラム名:MUSIC 2.M

88)

第2章楽全パート。音楽データ"DATA . M・1"を作成。

⑧リスト8(プログラム名:MUSIC 3.M

88)

第3章楽全パート。音楽データ"DATA A. M・3"を作成。

⑨リスト9(プログラム名:MUSIC 4.M

88)

第4章楽全パート。音楽データ"DATA A. M・4"を作成。

⑩リスト10(プログラム名:MUSIC 5.M

88)

第5章楽全パート。音楽データ"DATA A. M・5"を作成。

⑪リスト11(プログラム名:MUSIC 6.E

nd)

エンディング曲(ブランデンブルグ協奏曲第6番第2章楽)。音楽データ"DATA A. End"を作成。

⑫リスト12(プログラム名:MUSIC 7.O

pe)

オープニング曲(管弦楽組曲第2番第6章楽)。音楽データ"DATA . Ope"を作成。

⑬リスト13(プログラム名:list p1)

第1章楽演奏用プログラム。

⑭リスト14(プログラム名:list p2)

第1章以外の章楽の演奏用プログラム。

リスト1

10 ' [OVERTURE] Part of 1st Violin

20 Programmed by D.Kobayashi in June,1985

30 "save "MUSIC1.M8"

40 DATA "T7B04V10Q8#2L16"

50 DATA "04L16D4&DF#E", "B4&B>DC#B"

60 DATA "05L16G4&GEF#G", "C#4&C#A>C#"

70 DATA "04L16E4.E8", ">E8.C#A>G."

80 DATA "04L16F8.DF#8.A", ">D8.A>D8.E"

90 DATA "05L16F#4&F#GAF#", "B4&B#GAB"

100 DATA "05L16E4&EEF#G#", "A8.EC#<A"

110 DATA "05L32D4.D8", ">D8.EF#E8.D16"

120 DATA "05L16C#8.<A>C#8.E", "A8.G32#32G8.E"

130 DATA "05L32B4.D#8", ">E8.F#16F#8.EF#"

140 DATA "05L16G4&GF#ED", "C8.CD8.E"

150 DATA "04L16A#4.>G8", ">F#8.ED8.C8"

160 DATA "05L16D8.C#<B8.>C#", "C#4.<B8"

170 DATA "04L16B4&BAGF#", "A8.>F#<A8.G#"

180 DATA "04L16G#8.EG#8.A", ">E4.E8"

190 DATA "05L16E4&EAB>C#", "d#c#def#ef#g#g"

200 DATA "05L16AG8.F#E8", ">B4.D8"

210 DATA "05L8C#4.<B", "B4.A"

220 DATA "04L16A4A>C#<BA", ">G8.EF#8.G"

230 DATA "05L32A4.C8", ">C8.CD8.<A16"

240 DATA "04L16B4&BAGF#", "EF#GAB>C#D<B"

250 DATA "05L16C#<DC#<BAB>C#D", "EF#GAGF#GE"

260 DATA "05L16F#8.GAB.B", "C#8.DD8.C#32D32"

270 DATA "05L32E8.<A16>G8.(F#EF#)16", "F#4.E8"

280 DATA "05E1"

290 DATA "T10005L16E8<AB>C#8C#D", "E8C#DE8EF#"

300 DATA "05L16G8GF#GAG", "L8F#ADF#"

310 DATA "F#ACF#&.", "F#ADF#"

320 DATA "F#ABGDG#&.", "G#BEG#"

330 DATA "05L16AE8C#DEF#EF#", "GEKAB>C#DC#D"

340 DATA "05L16EDC#DEDCC#<B", "A>DC#<BAGF#E"

350 DATA "04L8D>DD", "DDDD"

360 DATA "05L8DEE", "EEEE"

370 DATA "05L8EEEE", "EL16<AB>C#8G#D"

380 DATA "05L16E8C#DE8EF#", "G8GF#GAG"

390 DATA "05F#8F#EF#8GF#", "E8EDEF#E"

400 DATA "05D8DEF#8F#G", "A8F#GAB"

410 DATA "06C8C#<C8D", "BGEF#GAF#G"

420 DATA "05AF#DEF#GEF#", "GF#GEF#GF#G"

430 DATA "05EAC#DEF#EF#", "GEAB>C#DC#D"

440 DATA "05EDC#DEF#EF#", "C#GF#GAF#DF#"

450 DATA "05GF#EF#GAB", "L8F#D#C#E"

460 DATA "05L16C#<AB>C#DEEF#G", "F#EDE<A8>C#8"

470 DATA "05D<AF#ADF#EG", "F#A>D<F#E>D<E>C#"

480 DATA "05D<A>DEF#A#G#B", "AEC#AA>B"

490 DATA "05AEC#DEF#EF#", "GEAB>C#DC#D"

500 DATA "05EDC#DEDCC#<B", "A8F#GF#EDCC#"

510 DATA "05DF#B>C#D#E", "F#AD#EF#GF#G"

520 DATA "05A<C#<C#BA", "B>C#DEF#GAF#"

530 DATA "05GED#E<A>G", "F#D<A>C#D>C#F#"

540 DATA "05EC#<C#G#BA#>C#<F#>E", "DC#DEF#GE"

550 DATA "04A#>EF#GF#EF#E", "<A#>EF#EF#EF#E"

560 DATA "04A#>EF#GF#EDC#C#", "<A>D#F#D#F#D#F#D#"

570 DATA "04A>D#F#D#F#D#F#D#", "<A>D#F#EDC#C#BA"

580 DATA "04G#>DEDEDEDC#<B", "G#DEDEDC#C"

590 DATA "04G#>DEDEDC#<B", "G#DEDEDC#C"

600 DATA "04G#>DEDEDC#<B", "G#DEDEDC#C"

610 DATA "04F#DEGF#GEF#", "AF#G#BA>C#D"

620 DATA "05C#<AB>D#C#DEF#", "E8.F#<B8.A"

630 DATA "04AB>C#DEF#EF#", "GEAB>C#DC#D"

640 DATA "05EDC#DEDCC#<B", "L8A>F#D#8"

650 DATA "D#F#C#A>D#8", "D#F#C#A>D#8"

660 DATA "05L16E8C#EF#G8G#A", "B8G#AB8B>C#"

```

670 DATA "05D8DC#D8ED", "C#8C#B>C#8DC#"
680 DATA "04B8BAB>C#>B", "A8AB>C#8C#D"
690 DATA "05L16E8C#DE8EF#", "G8GF#G8AG"
700 DATA "05F#D<B>C#DECD#", "EC#A#B>C#D<B>C#"
710 DATA "05DEC#DEDC#<B>", ">L8F#F#F#F#"
720 DATA "05L8F#F#F#F#", "F#F#F#F#L16"
730 DATA "05F#EF#GF#B#A#B#", "DEEF#EGF#E#"
740 DATA "05L8DF#G#B>", "L16A#F#G#A#B>C#D#E"
750 DATA "05FG#F#EDC#<B#>", "LBB>C#>F#A#>"
760 DATA "04L16BF#DF#B>DCE#", "DF#BDC#B#C#A#"
770 DATA "05L16BF#DF#B>D>G#>E#>", "F#C#<A#>F#<C#>F#<G#>E#>"
780 DATA "F#C#<A#>C#F#G#A#F#", "C#<B>C#<A#>C#<A#>"
790 DATA "04B#A>C#D#EF#G#>", "AF#D#F#<B>A>G#F#"
800 DATA "05GE<B>B>E#>A", "G#F#G#B#G#F#<G#>B"
810 DATA "05AEC#E<A#>C#D", "EF#GAG#F#GE"
820 DATA "F#D<A#>C#<B>D>G#F#>", "EC#<G#>A#<C#>F#>E#"
830 DATA "05D<B#>F#AGBE#D", "C#E#A#<D#>C#<A#>D"
840 DATA "05EG<A#>EF#ADF#>", "C#E#A#<C#>EG#F#E"
850 DATA "05F#D<B>D>G#>F#E#>", "C#D#C#<B>AG#F#E#"
860 DATA "04D#ABABABA", "D#ABABABA"
870 DATA "04D#AB>C#<B>AG#F#>", "D#B#B#D#G#B#G#"
880 DATA "04D#G#B#G#B#G#B#>", "D#G#B#G#F#E#D"
890 DATA "04C#GAGAGAG", "C#GAGAGAG"
900 DATA "04C#GABAG#E#>", "CF#A#F#A#F#A#"
910 DATA "04CF#A#F#A#F#A#", "CF#A#G#F#E#E#"
920 DATA "03B#G>C#<D#>C#E", "D#<B>C#E#F#E#G"
930 DATA "04F#DEFG#AGB", "ADF#A#<D#>E#>"
940 DATA "05C#8<A#>C#8C#D", "E8C#D#F#E#"
950 DATA "05L16G8GF#G8AG#>", "L8F#ADF#E#>"
960 DATA "F#ACF#E#>", "F#ADF#E#"
970 DATA "05L8BGD#G#>", "G#BEG#"
980 DATA "05L16AEC#DEF#E#>", "GE<A#>C#D#C#D#"
990 DATA "05L16EDC#DED#C#<B>", "A>D#C#<B>AG#E#"
1000 DATA "04L8D#DDO", "DDDD"
1010 DATA "05L8DEEE", "EEEE"
1020 DATA "05L8EEEE", "EL16<A#>C#8C#D"
1030 DATA "05L16E8C#DE8EF#>", "G8GF#G8AG"
1040 DATA "05F#8F#EF#8GF#>", "EEDE8F#E#"
1050 DATA "05D8DEF#8F#G", "A8F#G8A8B"
1060 DATA "06C8<B>C8DC", "<BGEF#GAF#G#"
1070 DATA "05AF#D#EF#GE#>", "GF#GF#E#D#F#"
1080 DATA "05EAC#DEF#E#F#>", "GE<A#>C#D#C#D"
1090 DATA "05EDC#DEGF#E", "AG#GAF#D#F#>"
1100 DATA "05G#F#EF#GBAG", "L8F#D#C#B>"
1110 DATA "05L16C#<A#>C#DEF#G", "F#E#D#<A#>C#8"
1120 DATA "T7805D48DEF#D", "<A#>A#<A#>C#<A#>"
1130 DATA "04F#48F#GAF#>", "D48D, L32F#GAB"
1140 DATA "05L16C8, CD8, E", "D8, C#B8E"
1150 DATA "04B4C4", "B4, C#B8"
1160 DATA "05D48DC#D#E", "D#4, E8"
1170 DATA "05F#48F#AGF#>", "B8, AG8, F#"
1180 DATA "05G8, F#E8, F#>", "F#4, E8"
1190 DATA "05E48EEDC#>", "D8, BE8, D"
1200 DATA "05C#8, <A#>C#8, E", "A4, A8"
1210 DATA "05A48A, L32<B>C#DEF#>", "G4, G8"
1220 DATA "05G48G16, L32F#EDC#<B>", "A8, GF#<G#>, >E16"
1230 DATA "04L16F48F#GAB", ">C4, D8"
1240 DATA "04B48AGF#>", "EF#GAB>C#D#B"
1250 DATA "05C#D#C#<B>B>C#D", "EF#GAG#F#GE"
1260 DATA "05F#8, (EDC#)16D8, <B>", ">#4, D8"
1270 DATA "T18005D8<A#>C#8C#D", "E8C#D#E#F#"
1280 DATA "05D1"
1290 OPEN "DATA A.M:1" TO OUTPUT AS #1
1300 RESTORE FOR I=0 TO 47:GOSUB *PLAY:NEXT I
1310 RESTORE 50:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1320 RESTORE 298:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1330 RESTORE 390:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1340 RESTORE 1280:GOSUB *PLAY
1350 CLOSE:END
1360 *PLAY
1370 READ A$:PRINT A$:WRITE #1,A$
1380 RETURN

```

リスト2

```

10 ' [ OVERTURE ] Part of 2nd violin
20 ' Programmed by D.Kobayashi in June,1985
30 ' save "MUSIC1.M8B"
40 DATA "T7804V1BQ8ZL16"
50 DATA "04L16R8B1GA#F#>", "F#8", "D8B, D8, B"
60 DATA "04B8, G8B, E", "D8", "E8B, C#>"
70 DATA "05C8<, <G#>, >G", "G8, EE8, C#>"
80 DATA "05D28", "D8, <F#>A8, >C#>"
90 DATA "05D8, F#F#>D", "D8, <B#>8, G#>"
100 DATA "04G#8, B8B, G#>", "E4A4"
110 DATA "04A8, G#A8, B", "E4B4"
120 DATA "04A8, EA8, B", ">C4, C8"
130 DATA "04D#A#F#>", "B8, B#<C#>, <B>"
140 DATA "L16B8, >E8, <B>", "G4, G8"
150 DATA "F#4, A#8", "B4, >C#>"
160 DATA "L16B8, A#B8, B", "A#4, B8"
170 DATA "L16B8, <B#>, >D#>", "F#4, F#8"
180 DATA "E8, >E8, D", "C#8, C#<B#>, <B>"
190 DATA "A4, A8", "A4, A8"
200 DATA "B8, AG#8, F#>", "E8, G#A8, B"
210 DATA "A4, A8", "G#4, E8"
220 DATA "L16B8, >E8, C#>", "C4A4AG#E#"
230 DATA "D4, <A#>", ">A8, GA8, F#>"
240 DATA "G8, F#G8, A", "B8, <D#>D, E"
250 DATA "<A8, EC#8, <A#>", ">A8, F#D8, A"
260 DATA "A4, G8", "E8, F#F#8, D"
270 DATA "A4, A8", "A8, >D32C#32D8, C#"
280 DATA "C#1"
290 DATA "T18005C4A4", ""
300 DATA "R2", "R804DEF#8F#G#"
310 DATA "A8F#GA8AB", ">C8C#D8C#D"
320 DATA "<B8GAB8B>C#>", "D8D8C#D8E"
330 DATA "L8C#E#A#<C#>", "C#E#G#C#"
340 DATA "C#E#A#<C#>", "L16D#AF#GABAB"
350 DATA ">C#<ADEF#GF#G#>", "05G#A#G#F#E#"
360 DATA "L8DBEG#>", "G#D8G#"
370 DATA "G#BEG#>", "L16A#<A#>D#EF#F#"
380 DATA "GE<A#>C#D#C#D", "EDC#DED#C#<B>"
390 DATA "A8F#B84", "B8A#B#>C#<B>G#"
400 DATA "F#>F#<B>C#DEDE", "F#ADEF#GF#G#"
410 DATA "AG#F#GAG#E#>", "D8B8E"
420 DATA "E8<A#>D4", "D8C#D8C#<B>D"
430 DATA "L8C#E#A#<C#>", "C#E#G#C#"
440 DATA "C#E#A#<C#>", "D#A#D4"
450 DATA "D#GE#C#>", "D#A#D"
460 DATA "L16EC#DE#A8>>C#8", "L8D#BAA"
470 DATA "A4R8A", "ABBA"
480 DATA "A4R8B", ">C#<A#>F#E#"
490 DATA "E4R8E", "EER8E"
500 DATA "C#C#R8C#>", "C#<A#>R8A#>"
510 DATA "B#F#R8B", ">C#C#R8"
520 DATA "<F#>A8A", "D#B8R>D#>"
530 DATA "E4R4", ""
540 DATA "..."
550 DATA "L16R8<F#>G#A#B#A#>", ">C#<A#>B>C#8C#D#"
560 DATA "E8DE8F#E", "D#4<B4"
570 DATA "R4>D#4", "R4F#4"
580 DATA "R8<EF#G8B8G#A", "B8G#AB8B>C#"
590 DATA "D8D#D8ED", "C#4<A4"
600 DATA "R4>C#4", "R4E4"
610 DATA "D4R4", ""
620 DATA "..."
630 DATA "R8L8E#A#<C#>", "C#E#G#C#"
640 DATA "C#E#A#<C#>", "L16D#C#DEF#GF#G#"
650 DATA "AF#<B>C#D#E#D#E", "F#E#D#EF#ED#C#"
660 DATA "L8<B>EEE", "EEEE"
670 DATA "EEE<B>", "L16AG#ABAG#F#A"
680 DATA "GAG#G#F#E#B8B#>", ">A#<C#>F#G#ABAB"
690 DATA ">C#E#A#<C#>C#D#C#D", "EDC#DED#C#<B>"
700 DATA "L8AF#B4", "B#<A#>F#A#"
710 DATA "L16F#GEF#G8E8", ">C#F#<A#>B>C#D#C#D"
720 DATA "EC#<F#>G#ABAB", ">C#<B>AB>C#EDC#D"
730 DATA "<B8F#B84", "L8BEC#A#"
740 DATA "BBBE", "C#4R8>C#"
750 DATA "<BA#F#F#>", "DGF#F#>"
760 DATA "F#4R8F#>", "F#GGF#>"
770 DATA "F#4R8C#>", "C#DDC#"
780 DATA "C#4R8C#>", "C#A8A8"
790 DATA "D#A8R8B", "B8R8B"
800 DATA "B#ER8E", "<B8R8B"
810 DATA ">C#ER8E", "C#<C#>R8C#"
820 DATA "DF#F#R8", "R8EER8"
830 DATA "R8D8R8", "R8E#A#F#"
840 DATA "C#E#A#>A2", "A2"
850 DATA "A#F#<B>G#>", "A4R4"
860 DATA "<A4>F#4", "R4D#4"
870 DATA "R4<B4", "R8L16EF#G#B8G#A"

```



```

880 DATA "B8G#AB8B>C#","D8C#D8B#<B"
890 DATA ">C#E4", "R4C#4"
900 DATA "R4<A4", "R8DEF8F#G"
910 DATA "A8F#GA8AB", ">C8C<B>C8<BA"
920 DATA "B4R4", ""
930 DATA ""
940 DATA "R8L8>E(A)<C#", "C#E6C#<B"
950 DATA "C#E(A)<C#", "L16D8<DEF#F#G"
960 DATA "A8F#GA8AB", ">C8C<B>C8C"
970 DATA "<B8GAB8B>C#", "D8C#D8ED"
980 DATA "L8C#E(A)<C#", "C#E6C#<B"
990 DATA "C#E(A)<C#", "L16D8<F#GABAB"
1000 DATA ">C<ADEF#GF#G", "AGF#GAGF#E"
1010 DATA "L8D8EG", "C#E8D8G"
1020 DATA "C#E6C#", "L16A<ACDEF#EF#F"
1030 DATA "GE<AB>C#D8C#D", "EDC#DED8C#<B"
1040 DATA "A8F#8B4", "B8A8#<C#<BA#G#H"
1050 DATA "F#>F#<B>C#D8E", "F#ADEF#GF#G"
1060 DATA "AGF#GAGF#E", "D8<B8>E4"
1070 DATA "E8<AB>D4", "D8C#D8C#<B>D"
1080 DATA "L8C#E(A)<C#", "C#E6C#<B"
1090 DATA "C#E(A)<C#", "C#E6C#<B"
1100 DATA "D<GE>C#", "D<AD>"
1110 DATA "L16E<C#D8E(A)>C#8", "L8D<BAA"
1120 DATA "T70L16A8.DD8.F#H", "F#8.AA8.F#H"
1130 DATA "D8.F#F#8.A", "A8.>C8C.<A"
1140 DATA "F#4.F#8", "G8.AG8.F#H"
1150 DATA "G8.BB8.G", "D4.E8"
1160 DATA "C#A.>DF#8.G", "A8.>C8C.<B"
1170 DATA "D8.F#D8.A", "<B8.>D8E8.F#H"
1180 DATA "E8.D8E8.F#H", "D4.E8"
1190 DATA "E8.<E8.G#H", "B4.B8"
1200 DATA "A8.CE8.A", ">C#4.A8"
1210 DATA "F#8.>DD8.C#H", "C#4.B8"
1220 DATA "A8.>C#C#8.E", "E8.D8E8.C#H"
1230 DATA "D8.<A8.F#H", "F#8.DD4"
1240 DATA "D8.F#G8.A", "F#8.>C8D8.E"
1250 DATA "C#A8.EC8.<A", ">A8.F#D8.A"
1260 DATA "A8.GA8.B", "A4A4"
1270 DATA "T100A4R4", ""
1280 DATA "A1"
1290 OPEN "DATA B.M-1" FOR OUTPUT AS #1
1300 RESTORE
1310 RESTORE:FOR I=0 TO 47:GOSUB *PLAY:NEXT I
1320 RESTORE 59:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1330 RESTORE 298:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1340 RESTORE 308:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1350 RESTORE 1208:GOSUB *PLAY
1360 CLOSE:END
1370 *PLAY
1380 READ BS:PRINT BS:WRITE #1,BS:RETURN

```

```

380 DATA "L8AAAA", "AAAA"
390 DATA "AL16ED8C#B>D", "C#D8C#BA#8>C#8"
400 DATA "L8DF#<B>D", "D>DDD"
410 DATA "DDC#A", "BBBB"
420 DATA "AAAA", "GAAA"
430 DATA "AAAA", "AAAA"
440 DATA "AAAA", "ADF#B"
450 DATA "B>C#16D16C#A", "AAGB"
460 DATA "AEAA", "L16AGF#GEF#GE"
470 DATA "F#4R8L8E", "DF#G"
480 DATA "F#4R8E", "EF#F#B"
490 DATA ">C#4R8C#", "C#C#R8<B"
500 DATA "A#A#R8A#", "F#F#R8F#"
510 DATA "F#R8F#", "A#A#R8"
520 DATA "D#F#R8F#", "F#F#R8B"
530 DATA "B4R4", ""
540 DATA ""
550 DATA "R4>C#4", "R4<A#4"
560 DATA "R4F#4", "R8L16C#D#D8D#E"
570 DATA "F#8D#EF#8F#G", "A8AG#A8BA"
580 DATA "G#4B4", "R4E4"
590 DATA "R8E4", "R8<AB>C#8C#D"
600 DATA "E8C#DE8EF#", "G8F#G8AG"
610 DATA "F#4R4L8", ""
620 DATA ""
630 DATA "R8AAA", "AAAA"
640 DATA "AAAA", "ABBB"
650 DATA "BBBB", "BBBBL16"
660 DATA "B>E<G#AB>C#<B>C#", "D<BEF#G#AG#A"
670 DATA "BAG#ABAGF#F", "E8C#R8F#A"
680 DATA "F#8E#F#G#F#E#D#", "L8C#C#<F#A"
690 DATA "A>AAA", "AAAE"
700 DATA "F#F#F#F#F#", "EGF#F#H"
710 DATA "BF#BB", "A#>C#<F#A"
720 DATA "A>C#<E#B", "A#>C#<F#A#H"
730 DATA "B#D>F#H", "G#A#F#H"
740 DATA "F#D<EB", ">F#4R8A#"
750 DATA "F#C#<B>D", "C#<EDC#"
760 DATA "D4R8C", "C#<D#C#C#"
770 DATA "D4R8G", "AF#BB"
780 DATA "A4R8F#", "F#F#R8F#F#"
790 DATA "F#F#R8F#", "D#F#R8D#H"
800 DATA "EGR8B", "EER8E"
810 DATA "EAB8A", "AER8A"
820 DATA "AR8R8G", "GR8R8F#"
830 DATA "F#R8R8E", "EGF#D"
840 DATA "AC#D#F", "EGC#A"
850 DATA ">D2", "E4R4<L16"
860 DATA "R8B>C#D#8D#E", "F#8DEF#8F#G"
870 DATA "A8AG#A8BA", "G4E4"
880 DATA "R4G#4", "R4B4"
890 DATA "R8<AB>C#8C#D", "E8C#DE8EF#"
900 DATA "G8GF#G8AG", "F#4D4"
910 DATA "R4F#4", "R4A4"
920 DATA "D4R4", ""
930 DATA ""
940 DATA "R8C#DEF#EF#H", "GE<AB>C#D8C#D"
950 DATA "EDC#DED8C#<B", "A>AF#GABAB"
960 DATA ">C<ADEF#GF#G", "AGF#GAGF#E"
970 DATA "D<B>EF#G#AG#A", "F#F#R8F#"
980 DATA "E8<AB>C#8C#D", "E8C#DE8EF#"
990 DATA "G8GF#G8AG", "L8F#ADF#"
1000 DATA "F#ACF#F", "F#ADF#L16"
1010 DATA "G8G#AB>C#<B>C#", "D<BEF#G#AG#A"
1020 DATA "BAG#ABAGF#F", "L8EAAA"
1030 DATA "AAAA", "AAAA"
1040 DATA "AL16ED8C#B>D", "C#D8C#BL8A#>C#>C#"
1050 DATA "DDC#<B>D", "D>DDD"
1060 DATA "D#<A", "BBBB"
1070 DATA "AAAA", "AAAA"
1080 DATA "AAAA", "AAAA"
1090 DATA "AAAA", "ADF#B"
1100 DATA "B>C#16D16C#A", "AAGB"
1110 DATA "AEAA", "L16AGF#GEF#GE"
1120 DATA "T70F#4R8R16D16", "L16D8.F#F#8.A"
1130 DATA "A8.<A8.8D", "F#8.AA8.F#H"
1140 DATA "D8.<A8.8C", "D8.D8D8.F"
1150 DATA "D4.D8", "G8.BB8.G"
1160 DATA "A4.G8", "F#8.EF#8.G"
1170 DATA "F#8.AA8.8C", "F#8.<B>E8.B"
1180 DATA "B8.AB8.8C", "C#4.B8"
1190 DATA "B2", "R8R16E8E.G#H"
1200 DATA "E8.AA8.G", "F#8.F#F#8.E"
1210 DATA "D8.<B>F#8.D", "D8.<A>E8.E"
1220 DATA "E8.AA8.E", "A4.A8"
1230 DATA "A8.F#F#8.D", "<A8.>DF#8.A"
1240 DATA "G8.D8.F#H", "G8.EF#8.G"
1250 DATA "E8.AA8.E", "E8.C#D8.E"
1260 DATA "<A8.>ED8.G", "E4.F#8"
1270 DATA "T100F#4R4", ""
1280 DATA "F#1"
1290 OPEN "DATA C.M-1" FOR OUTPUT AS #1
1300 RESTORE
1310 RESTORE:FOR I=0 TO 47:GOSUB *PLAY:NEXT I

```

リスト3

```

10 * [ OVERTURE ] Part of Viola
20 * Programmed by D.Kobayashi in June,1986
30 save "MUSIC1.M8C"
40 DATA "T7004V7QBL16"
50 DATA "R4R8R16A", "A8.F#F#8.D"
60 DATA "D8.BB8.G", "G8.GG8.E"
70 DATA "G8.C#C#8.E", "<A8.>AA8.A"
80 DATA "A2", "A8.DF#8.G"
90 DATA "A8.>DD8.<A", "F#8.DD8.<B"
100 DATA "B8.G#G#8.B", ">E4F#4"
110 DATA "F#8.DD8.C#H", "C#4E4"
120 DATA "E8.C#E8.G", "F#8.D#E8.G"
130 DATA "F#4B4", "B8.EE8.D#H"
140 DATA "E8.BB8.G", "E4.D8"
150 DATA "C#4.C#8", "D8.EF#4"
160 DATA "F#8.EF#8.G", "F#4F#4"
170 DATA "F#2", "R8R16<BB8.>D#H"
180 DATA "<B8.G#B8.>E", "F#4.E8"
190 DATA "C#8.AA8.G", "F#8.F#F#8.E"
200 DATA "D2", "D8.DC#8.<B"
210 DATA ">E8.D8.F#H", "E4D4"
220 DATA "C#2", "C#8.C#D8.E"
230 DATA "F#8.AA8.F#H", "F#4D4"
240 DATA "D8.D8.F#H", "G8.EF#8.G"
250 DATA "E8.AA8.E", "E8.C#D8.E"
260 DATA "D4.D8", "A8.AA8.F#H"
270 DATA "E8.D8.<A", ">A4A4"
280 DATA "A1"
290 DATA "T100O4A4R4", ""
300 DATA ""
310 DATA ""
320 DATA ""
330 DATA "R8<AB>C#8C#D", "E8C#DE8EF#"
340 DATA "G8GF#G8AG", "L8F#ADF#"
350 DATA "F#ACF#F", "F#ADF#"
360 DATA "L16G#AB>C#<B>C#", "D8<EF#G#AG#A"
370 DATA "BAG#ABAGF#F", "E4R8A8"

```

```

1320 RESTORE 50:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1330 RESTORE 290:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1340 RESTORE 300:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1350 RESTORE 1280:GOSUB *PLAY
1360 CLOSE:END
1370 *PLAY
1380 READ C$:PRINT C$:WRITE #1,C$:RETURN

```

リスト4

```

10 ' [Overture] Part of Harpsichord
20 ' Programmed by D.Kobayashi in June,1985
30 ' save "MUSIC1.M8D"
40 DATA "T7804V6G800L16"
50 DATA "04D2",""
60 DATA "",""
70 DATA "R4DF#ED","C#8.EC#8.<A"
80 DATA ">D2","RDC#BAGF#E"
90 DATA "D2",">D2"
100 DATA "R8.DC#8.<B",">C#4F#4"
110 DATA "<B>C#DEF#G#A#F","G#4.F#G#"
120 DATA "A2",""
130 DATA "RGF#ED#C#BA","G8.GA8.B"
140 DATA "E2",">E2"
150 DATA "R8.GF#8.E","D8.C#<B8.A#"
160 DATA "B8.>C#D8.E","F#8.E32D32E8.F#"
170 DATA "D#2","RCK<B>C#D#C#D#C#B"
180 DATA "E2","RDC#DEDEC#"
190 DATA "F#2","RGF#EDED#C#"
200 DATA "<B4&B8>C#D","<C#8.EF#8.G#"
210 DATA "A8.B>C#8.D","E4E4"
220 DATA "A2","R4RBAG"
230 DATA "F#4&F#G#E","D8.EF#8.D"
240 DATA "G2",">G2"
250 DATA "R4RF#ED","C#8.AB8.>C#"
260 DATA "D8.EF#8.G","R4BGF#GAB"
270 DATA ">C#<B4B>C#DECE#","D4C4"
280 DATA "<A8.7A8.EE8.C#C#8">
290 DATA "T10003A4R4",""
300 DATA "",""
310 DATA "",""
320 DATA "",""
330 DATA "",""
340 DATA "","R804DEF#8F#G"
350 DATA "A8#G#GA8A8",">C8C<B>C8DC"
360 DATA "<B8EF#G#8G#A","B8G#AB8B>C#"
370 DATA "D8DCD8ED",">C#8E<A>C#8"
380 DATA "L8C#E<G>C#","C#E<A>C#8"
390 DATA "<C>D<GB>","C#>C#<F#A#"
400 DATA "<C>BAG","F#BAG"
410 DATA "F#ADA","<G>G#E"
420 DATA "<C#F#C#B>D","<E4D<D"
430 DATA "A116A8>C#8C#D","E8C#DE8EF#"
440 DATA "G8GF#G8AG",">F#8F#EF#8GF#"
450 DATA "E8EDE8F#E","DEF#DGAF#G"
460 DATA "L8AGF#E","DGA<A"
470 DATA ">D4R8C#8","D<BGA"
480 DATA "D4R8D",">C#F#4E"
490 DATA "<A>AAA","R8GG"
500 DATA "R8F#F#F#","R<C#F#F#F#"
510 DATA "R8BBB","R8D>F#F#F#"
520 DATA "R8D#D#D#D","R8<BBB"
530 DATA ">E>E<A>C#","<D>C#D"
540 DATA "C#>C#<F#A#","<B>BGE"
550 DATA "F#4R4","<F#4R4"
560 DATA "R8F#A#F#","B4R4"
570 DATA "<B4R4","R8>C#B>D#"
580 DATA "E4R4","<E4R4"
590 DATA "R8EG#E","A4R4"
600 DATA "<A4R4","R8>A<A>C#"
610 DATA "DC#DE",">F#EF#G#"
620 DATA "AG#A#F#","C#DE<E"
630 DATA "A116A8>C#8C#D","E8C#DE8EF#"
640 DATA "G8GF#G8AG",">F#8C#D#8D#E"
650 DATA "F#8DEF#8F#G","A8AG#AB8A"
660 DATA "L8B#BEG#","G8BDG"
670 DATA "G#BEG#","<A>ADF#"
680 DATA "<G#>G#C#E#","<F#F#ED"
690 DATA "C#F#ED","C#E<A>C#"
700 DATA ">D>C#B","C#>C#F#A#"
710 DATA "C#BEG","F#116F#G#A8A8B#"
720 DATA "<C#8A#B>C#8C#D","E8EDE8F#E"
730 DATA "D8DCD8ED",">C#8C<B>C#8DC"
740 DATA "<C#C#D#B>E#G#","F#4R8E"
750 DATA "L8DF#B#F#","GEF#<F#"
760 DATA "B4R8A#",">BGEF#"
770 DATA "<B4R8>B",">A>D<B>C#"
780 DATA "<F#F#F#F#","R8EE"
790 DATA "R8D#D#D#D","R8<BBB"
800 DATA "R8EE","R8DD"
810 DATA "R8C#C#C#","R8A<A"
820 DATA ">D>C#B","<C#>C#<F#A#"

```

```

830 DATA "<B>BEG","<A2"
840 DATA "",""
850 DATA "",">RB>C#<A"
860 DATA "F#4R4","B4R4"
870 DATA "R8B>D#<B>",">E4R4"
880 DATA "<E4R4","R8E>E#G#"
890 DATA "A4R4",">A4R4"
900 DATA "R8A<C#<A>",">D4R4"
910 DATA "<D4R4","R8>D<DF#"
920 DATA "GF#GA","B4B>C#"
930 DATA "DC#DE",">F#EF#G"
940 DATA "AAAA","AAAA"
950 DATA "AEC#<A>",">D>DDD"
960 DATA "DDDD",">D<A#D"
970 DATA "G#EE","E<BGE"
980 DATA "A4R4",">A4R4"
990 DATA ">A4R4","R8L16EF#8F#G"
1000 DATA "A8F#GA8A8",">C8C<B>C8DC"
1010 DATA "<B8EF#G#8G#A","B8G#AB8B>C#"
1020 DATA "D8DCD8ED","L8C#E<A>C#"
1030 DATA "R8G>C#","RE<A>C#"
1040 DATA "<C>D<GB>","C#>C#<F#A#"
1050 DATA "<B>BAG","F#BAG"
1060 DATA "F#ADF#","<G>G#E"
1070 DATA "<C#F#F#<B>D","<E4D<D"
1080 DATA "L16A8A8>C#8C#D","E8C#DE8EF#"
1090 DATA "G8GF#G8AG",">F#8F#EF#8GF#"
1100 DATA "E8EDE8F#E","DEF#DGAF#G"
1110 DATA "L8AGF#E","DGA<A"
1120 DATA "T7804D2",""
1130 DATA "",""
1140 DATA "R4L16R<C#BA","B8.>C#D8.<D"
1150 DATA "G2",">G2"
1160 DATA "F#2","R8ABAGF#E"
1170 DATA "D#2","R8.<B>C#8.D#"
1180 DATA "E8.F#8A.>","B8.A32G#32A8.B"
1190 DATA "G#2","R8F#EF#G#F#G#E"
1200 DATA "A2","RGF#GAGAF#"
1210 DATA "B2","R<C#BAGF#ED"
1220 DATA "C#2","RDC#B4C#C#<A"
1230 DATA ">D2","REDC<BAGF#"
1240 DATA "G2",""
1250 DATA "R4RF#ED","C#8.AB8.>C#"
1260 DATA "D8.EF#8.G","A4C4"
1270 DATA "T100D4R4",""
1280 DATA "D1"
1290 OPEN "DATA D.M-1" FOR OUTPUT AS #1
1300 RESTORE
1310 RESTORE:FOR I=0 TO 47:GOSUB *PLAY:NEXT I
1320 RESTORE 50:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1330 RESTORE 290:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1340 RESTORE 300:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1350 RESTORE 1280:GOSUB *PLAY
1360 CLOSE:END
1370 *PLAY
1380 READ D$:PRINT D$:WRITE #1,D$:RETURN

```

リスト5

```

10 ' [Overture] Part of 1st Trombone
20 ' Programmed by D.Kobayashi in June,1985
30 ' save "MUSIC1.M8E"
40 DATA "T7804V7Q002L16"
50 DATA "05L8A4","A4"
60 DATA "B4","G4L16"
70 DATA "R8R16GG8.E","E8.AA8.E"
80 DATA "B8.AA8.F#","F#8.DD8.<A"
90 DATA "A8.AA8.F#","D8"
100 DATA "05E4L16EEF#G#","A8.EC#8.<A"
110 DATA "05L32D4.D8","D8.EF#E8.D16"
120 DATA "05L16C#8.<A>C#8.E","A8.G32F#32G8.E"
130 DATA "05L32B4.D8#","E8.F#16F#8.EF#"
140 DATA "05L16G4&GF#ED","C#D8.E"
150 DATA "04L16A#4.>B8",">F#8.ED8.C#"
160 DATA "05L16D8.C#<B8.>C#","C#4.<B8"
170 DATA "04L16B4&BBG#","A8.F#<A8.G#"
180 DATA "04L16C#8.EG#8A",">E4.E8"
190 DATA "05L16E4&E<A>B>C#",">C#def#ef#G#"
200 DATA "05L16A8#8.F#E8.<A","B4.D8"
210 DATA "05L8C#4.<B>","B4.A"
220 DATA "R8R16L1606C#8C#8.<A","A8."
230 DATA "R8R16>D8.<A>","A8."
240 DATA "04L16B4&BAG#",">E#GAB>C#D<B"
250 DATA "05R8R16EE8.A","A8.AG8.A"
260 DATA "F#8.GA8.E","E8.AD8.<B"
270 DATA "<A8.>A8.A","F#8.A>D8.<B"
280 DATA ">C#2"
290 DATA "T100V806C#4","V8"
300 DATA "05L16GGF#G8AG","L8F#ADF#8"
310 DATA "05L8R ACF#8","F#ADF#"
320 DATA "05L8GBDG#8",">G#BEG#"
330 DATA "05L16AEC#DEF#F#","GE<A>C#D#C#"

```

```

348 DATA "05L16EDC#DEDC<B>","A>D<C#BAGF#E"
350 DATA "V805L8R8DDD","DF#A#A"
360 DATA "G","R8EEE"
370 DATA "EG#GB","A"
380 DATA "V6L16E8C#DE8EF#","G8GF#G8AG"
390 DATA "05F#8F#EF#8GF#","E8EDE8F#E"
400 DATA "V8R8L16DEF#8F#G","A8F#GA8AB"
410 DATA ">C8CB#C8DC","L8<R8A"
420 DATA "05AR4A","L16GF#EGF#EDF#"
430 DATA "05E8","R8<L8AAA"
440 DATA "A>EEA","AR4L16BA"
450 DATA "G8R4AG","L8F#ABG"
460 DATA ">C#(A>D<G)","F#R8E"
470 DATA "D",""
480 DATA "",""
490 DATA "",""
500 DATA "",""
510 DATA "",""
520 DATA "",""
530 DATA "",""
540 DATA "",""
550 DATA "V6R8L804Q5F#F#F#","F#F#F#F#"
560 DATA "F#F#F#F#","Q8F#"
570 DATA ">D#2",""
580 DATA "D2",""
590 DATA "D2","C#<AAA"
600 DATA "AAAA","A>EAE"
610 DATA "F#",""
620 DATA "",""
630 DATA "R8VB8AE","G16E16EGE"
640 DATA "L16EDEDE8F#","L8D#FAF#"
650 DATA "D#F#AF#","L16F#EF#EF#8GF#F#"
660 DATA "L8EEEE","EEEE"
670 DATA "EBGB",">C#L16"
680 DATA "V604B8BAB8<C#B>","A8AB>C#8C#D"
690 DATA "05L16E8C#DE8EF#","G8GF#G8AG"
700 DATA "05F#D<B>C#DEC#D","EC#A#B>C#D<B>C#"
710 DATA "05DEC#DEDC<B>",">L8F#Q5F#F#F#"
720 DATA "05L8F#F#F#F#","F#F#F#F#"
730 DATA "05Q8B8",""
740 DATA "05L8DF#G<B>","L16A#F#G#A#B>C#D#E"
750 DATA "05F#GF#EDC#<BA#>","L8B>C#<F#A#>"
760 DATA "04L16B#F#DF#B>DCE","D#B#DCC#B#C#A#"
770 DATA "",""
780 DATA "",""
790 DATA "",""
800 DATA "",""
810 DATA "","05C#8C#8R8G8"
820 DATA "L8F#R4F#","ER4E"
830 DATA "ER4E","ER8ER8"
840 DATA "F#R8ER8","DR8GR8"
850 DATA "F#R8GR8","A<L16AAA8A"
860 DATA "A8",""
870 DATA "",""
880 DATA "",""
890 DATA "L8R8AAA","AAAA"
900 DATA "A>EEA","F#<F#F#A"
910 DATA "A>DDF#","F#AA>D"
920 DATA "D",""
930 DATA "",""
940 DATA "R4<AR8","AR8ER8"
950 DATA "ER8GR8","F#R4"
960 DATA "R8<DDD","D>A>D<A"
970 DATA "B",""
980 DATA "",""
990 DATA "",""
1000 DATA "R8DDD","DF#FA#"
1010 DATA "G","R8EEE"
1020 DATA "EG#GB","A"
1030 DATA "",""
1040 DATA "",""
1050 DATA "L1605R8DEF#8F#G","A8F#GA8AB"
1060 DATA "06C8C<B>C8DC","<B8R4B8"
1070 DATA "05A8R4A8","GF#EGF#EDF#"
1080 DATA "05E8","L8<AAA"
1090 DATA "A>EEA","AR4B16A16"
1100 DATA "GR4A16G16","F#ABG"
1110 DATA ">C#(A>D<G)","F#R8ER8"
1120 DATA "T78V805D4","F#4"
1130 DATA "A4",">D4"
1140 DATA "<R8R16A16A8.A16>","L16>D8.<AB8.>C"
1150 DATA "04B4",""
1160 DATA "V505D4DC#DE","D#4.E8"
1170 DATA "05F#48F#AGF#","D8.AG8.F#"
1180 DATA "05G8.F#E8.F#","F#4.E8"
1190 DATA "05E48EDC#C","D8.BE8.D"
1200 DATA "05C#8.<A>C#8.E","A4.A8"
1210 DATA "05A48A.L32<B>C#DEF#","G4.G8"
1220 DATA "05G4G16.L32F#EDC<B>","A8.GF#G8.>E16"
1230 DATA "R8R16L16DD8.F#","F#8.AA8.B"
1240 DATA "B8","G8"
1250 DATA "R8R16EE8.A","A8.BG8.A"
1260 DATA "D8.<A>D8.<A>","A8.F#A8.A"
1270 DATA "T180V804F#4",""

```

```

1280 DATA "04F#1"
1290 OPEN "DATA.E.M-1" FOR OUTPUT AS #1
1300 RESTORE:FOR I=0 TO 47:GOSUB *PLAY:NEXT I
1310 RESTORE 50:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1320 RESTORE 290:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1330 RESTORE 300:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1340 RESTORE 1280:GOSUB *PLAY
1350 CLOSE:END
1360 *PLAY
1370 READ E$:WRITE #1,E$:PRINT E$:RETURN

```

リスト6

```

10 ' [OVERTURE] Part of 2nd trombone
20 ' Programmed by D.Kobayashi in June,1985
30 save "MUSIC1.M8"
40 DATA "T7804V7Q8L16"
50 DATA "05L8F#4","F#4"
60 DATA "G4","E4"
70 DATA "R8L16EE8.A","A8.EE8.E"
80 DATA "D8.F#F#8.D","D8.<AA8.F#"
90 DATA "F#8.AA8.F#","D8"
100 DATA "",""
110 DATA "",""
120 DATA "",""
130 DATA "",""
140 DATA "",""
150 DATA "",""
160 DATA "",""
170 DATA "",""
180 DATA "",""
190 DATA "",""
200 DATA "",""
210 DATA "",""
220 DATA "R8.05L16AA8.E","E8"
230 DATA "R8.F#F#8.D","D8"
240 DATA "",""
250 DATA "R8.<AA8.>E","E8.F#F#8.E"
260 DATA "<A8>","A8"
270 DATA "R8R16>EE8.E","D8.F#A8.G#"
280 DATA "A2",""
290 DATA "T180V605A4",""
300 DATA "",""
310 DATA "",""
320 DATA "",""
330 DATA "",""
340 DATA "",""
350 DATA "","L8DDDD"
360 DATA "D",""
370 DATA "EEEE","E"
380 DATA "",""
390 DATA "",""
400 DATA "04R8DDD","DDDD"
410 DATA "L16D8>F#GA8BA","G8R4G8"
420 DATA "F#8R4F#8","EAAA"
430 DATA "A8",""
440 DATA "R8<L8AA>E","F#L16AGF#8D8"
450 DATA "E8GF#E8<A8>","F#DDE"
460 DATA "<A>EL16F#EF#G","L8AR8G"
470 DATA "F#",""
480 DATA "",""
490 DATA "",""
500 DATA "",""
510 DATA "",""
520 DATA "",""
530 DATA "",""
540 DATA "",""
550 DATA "","R804L8Q5V5F#F#F#"
560 DATA "F#F#F#F#","F#"
570 DATA "V6Q8",""
580 DATA "",""
590 DATA "",""
600 DATA "R8AAA","AA>ECA"
610 DATA "A",""
620 DATA "",""
630 DATA "R4>E","GER8"
640 DATA "<A>AAA","AR4>F#"
650 DATA "AF#R8F#","DF#R4"
660 DATA "","REEE"
670 DATA "EG#GB#","E"
680 DATA "",""
690 DATA "",""
700 DATA "",""
710 DATA "",""
720 DATA "Q5R8F#F#F#","F#F#F#F#"
730 DATA "Q8F#",""
740 DATA "",""
750 DATA "",""
760 DATA "",""
770 DATA "",""
780 DATA "",""

```



```

790 DATA "",""
800 DATA "",""
810 DATA "",""
820 DATA "",""
830 DATA "","AR8D"
840 DATA "ER8D","AR8E"
850 DATA "DR8D","EL16<AAA8A8"
860 DATA "L8A",""
870 DATA "",""
880 DATA "",""
890 DATA "","R8AAA"
900 DATA "AAA>E","<D<DDF#"
910 DATA "F#AA>D","DF#F#A#"
920 DATA "B",""
930 DATA "",""
940 DATA "R4E","ER8A"
950 DATA "AR8E","DR4D"
960 DATA "DDDD","D>F#DF#"
970 DATA "G",""
980 DATA "",""
990 DATA "",""
1000 DATA "","R8DDD"
1010 DATA "D",""
1020 DATA "R8EEE","E"
1030 DATA "",""
1040 DATA "",""
1050 DATA "R8<DDD","DDDD"
1060 DATA "D>L16F#GA8BA","L8GR4G"
1070 DATA "F#R4F#","EAAA"
1080 DATA "A",""
1090 DATA "R8<AA>E","F#L16AGL8F#D"
1100 DATA "EL16GF#L8E<A","A>DDE"
1110 DATA "<A>EL16F#EF#G","L8AR8G"
1120 DATA "T70V805F#4","D4"
1130 DATA "F#4","A4"
1140 DATA "R8.L16F#F#8.A","F#8.F#F#8.G"
1150 DATA "F#4",""
1160 DATA "",""
1170 DATA "",""
1180 DATA "",""
1190 DATA "",""
1200 DATA "",""
1210 DATA "",""
1220 DATA "",""
1230 DATA "R8.<AA8.>D","D8.F#F#8.G"
1240 DATA "G8","B8"
1250 DATA "R8.<AA8.>E","E8.F#F#8.E"
1260 DATA "D8.<A>D8.<A","A8.F#A8.A"
1270 DATA "T100V604F#4",""
1280 DATA "O4F#1"
1290 OPEN "DATA F.M-1" FOR OUTPUT AS #1
1300 RESTORE:FOR I=0 TO 47:GOSUB *PLAY:NEXT I
1310 RESTORE 50:FOR I=1 TO 46:GOSUB *PLAY:NEXT I
1320 RESTORE 290:FOR I=1 TO 198:GOSUB *PLAY:NEXT I
1330 RESTORE 300:FOR I=1 TO 194:GOSUB *PLAY:NEXT I
1340 RESTORE 1280:GOSUB *PLAY
1350 CLOSE:END
1360 *PLAY
1370 READ F$:WRITE #1,F$:PRINT F$:RETURN

```

リスト7

```

10 [ AIR ]
20 Programmed by D.Kobayashi in 1985
30 save "MUSIC2.M88"
40 DIM A$(17),B$(17),C$(17),D$(17)
50 A$(0)="T3204V7#2":B$(0)="T3204V70#2":C$(0)="T3204V604":D$(0)="T3203V5L8#24Q8"
60 A$(1)=">F#16AGL16GF#E#E32.DC#D4<B16A8.>A2#":B$(1)="D18D4BA2A8L16<c>c<b>8ac"
70 C$(1)="GA2<B4>E4E2L8ED#D#E":D$(1)="D>DC<C<B>BA<AG>GG<G#A>AG<GF#>F#E<"
80 A$(2)="L16AF#C<B>ED#AGG2<GE<BA>DC#GF#":B$(2)="<BRRR4L16B8>EDEF#GE<A8R8R4"
90 C$(2)="F#FR8R4E<BB>EER8R4":D$(2)="D#>D<B>B<E>ED<DC>C#<A>"
100 A$(3)="P#4.G#AD8D32E32F#>F#EED":B$(3)="A2A8G#16A16B8G#8"
110 C$(3)="C#4.DEC<A>D#":D$(3)="D>DC<C<B>BG#E"
120 A$(4)="L16C<BB32<C#32D8D8C#<BA2":B$(4)="L8A8AG#E2"
130 C$(4)="EF#C<B>EC#2":D$(4)="ADE<EL16B8AB>C#DGEF#EL846"
140 A$(5)="O5C#4C#16L32DC#<B>C<A16>A4.C8":B$(5)="O4A2&A16L16B>C8C<BAG"
150 C$(5)="O4E2&ED#16E16F#4":D$(5)="O3<A>AG<GF#>F#E<"
160 A$(6)="L8<B>B8L16L16AGF#G4G32L3F#EDC#16<B16":B$(6)="F#4.>D#E2"
170 C$(6)="L16F#GAF#D#BB4<B4":D$(6)="D#>D#F#<B#>E>ED<D"
180 A$(7)="L16A#B>C#8C#D8E8&F#G8&G8F#8":B$(7)="E2&E16L16DC#<BA#B>C#8"
190 C$(7)=">L16C#DEF#GF#GEL8F#E16D16C#F#":D$(7)="C#>C#<B<BA#B>C#<A#8"
200 A$(8)="L16EDC#<B>C#D32E32D8<B2":B$(8)="<L8BBBA#F#2"
210 C$(8)="L16F#EDG8F#ED2":D$(8)="L8B>G8F#<B>A<A"
220 A$(9)="L16>D4&DF#ED8A.A16G#16":B$(9)="E4F#4L16<B8>EF#G#AB8#"
230 C$(9)="L16<B8>B8AG#A8G8.F#E4":D$(9)="G#>G#F#<F#E>ED<D"
240 A$(10)="L16F#64E32.A<A8B8.C#32D32C#8.<BA4":B$(10)="O4L8BA4G#A2#"
250 C$(10)="L8EEF#EL16E8.DC#DEC#":D$(10)="C#<C#DE<A>AG<C"
260 A$(11)=">D4.F#16E16E4.G16F#16":B$(11)="A8L16B<C>B>C#D8D8D8C#<B>C#D#E8#"
270 C$(11)="<A8>D4<B8&L8B>E4C#4":D$(11)="F#>F#G<G#>G#A<A"
280 A$(12)="F#4.A16G16C":B$(12)="L16EDC#DGEF#8F#D#E#E4#"
290 C$(12)="C#F#4D#L16<B4&B>BGE":D$(12)="A#>A#B<B>E>ED<D"
300 A$(13)="<A4&L16A>C#EGGEF#8F#8F#G32A32":B$(13)="L16EC#EA<C#8A8&A8C#D<D4&

```

```

318 C$(13)="L8AGF#ED4A48":D$(13)="C#<C#<A>C#D<C>C"
320 AS$(14)="D48L16DF#A>C<B4.D8":B$(14)="D8E8F#4G28"
330 C$(15)="A8G8A4D2":D$(14)="<B<B4>AG<GF#>F#>"
340 AS$(15)="L16C#EG4<B8AB16>EF#32G32G#8E":B$(15)="G8B8>E4&L16EDC#<BA8B8"
350 C$(16)="L16E8<B>EGBAGF#L8EA&AG":D$(15)="<E<ED>DC#<A>DG"
360 AS$(16)="D32C#32<B8>C#16D8C#16D16D2":B$(16)="A4G16F#16G8F#2"
370 C$(16)="F#4E8A8A2":D$(16)="AGA<AQ8D2Q6"
380 AS$(17)="D32C#32<B8>C#16D8C#16D16D2":B$(17)="A4G16F#16G8F#2."
390 C$(17)="F#4E8A8A2":D$(17)="AGA<AQ8D2."
400 OPEN "DATA.M-2" FOR OUTPUT AS #1
410 FOR I=0 TO 4:GOSUB *PLAY:NEXT I
420 FOR I=1 TO 16:GOSUB *PLAY:NEXT I
430 FOR I=5 TO 15:GOSUB *PLAY:NEXT I
440 FOR I=17 TO 17:GOSUB *PLAY:NEXT I
450 CLOSE:END
460 *PLAY
470 WRITE #1,A$(1),B$(1),D$(1),C$(1):RETURN

```

リスト8

```

10 ' [GAVOTTE]
20 ' Programed by D.Kobayashi in 1985
40 DIM AS$(62),BS$(62),C$(62),D$(62),E$(62),F$(62)
50 AS$(0)="T140V0902":BS$(0)="T140V0902":C$(0)="T140V09"
60 D$(0)="T140V0700":E$(0)="T140V11":F$(0)="T140V11"
70 AS$(1)="04A4L8AF#":AS$(2)="G2<A4>GE":AS$(3)="F#2<A4>EC#":BS$(1)="04L8F#4F#D":BS$(2)="E2<E4>EC#":BS$(3)="D2<L4DE"
80 AS$(4)="D4C#CBABG4":AS$(5)="F#4EDA4AF#":BS$(4)="<A>DDC#":BS$(5)="<A2>F#L8F#D"
90 AS$(6)="G2<A4>GE":AS$(7)="F#2G#4BG#":BS$(6)="L8E2<E4>EC#":BS$(7)="D2D4<B4"
100 AS$(8)="05A4GF#EF#D4":AS$(9)="C#4F#DE4DC#":BS$(8)=">L8E4DC#<B4G#4":BS$(9)="L4AAAA"
110 AS$(10)="C#C#A4E4G#4":AS$(11)="A2":BS$(10)="04L4F#F#E":BS$(11)="E2"
120 AS$(12)="05L8E4EG#":AS$(13)="F#2>D4CF#A":BS$(12)="04L8E4E4":BS$(13)="E4<F#EF#4>#4"
130 AS$(14)="G2<C#4>A#<C#":AS$(15)="<B4>C#D<C#<B>G4":BS$(14)="F#4<GF#L4GF#":BS$(15)="BA#BB"
140 AS$(16)="05L8G4F#4<F#4F#D":AS$(17)="E2<F#4>EC#":BS$(16)="04B4A#4D4>D4":BS$(17)="D4L8C#<B>C#4A#4"
150 AS$(18)="05D4<B>D<C#<AB#4":AS$(19)=">F#4GF#EDC#D":BS$(18)="04B4L4DEF#8G8":BS$(19)="F#BBA#>"
160 AS$(20)="04B2B4L8BG#":AS$(21)="05A2<B4>AF#":BS$(20)="04F#2G4>D8<B8":BS$(21)="05C2<F#4>F#8D#8"
170 AS$(22)="G2C#4GE":AS$(23)="F#4EDEC#D4":BS$(22)="E2L404E#":BS$(23)="D<F#BB"
180 AS$(24)="C#404BA>A4<C#":AS$(25)="<B2>G4<B>D":BS$(24)="A2A#A":BS$(25)="05G4<GF#8L4GE"
190 AS$(26)="C#4B<C#4A>GE":AS$(27)="F#EDEC4A>C#4A":AS$(28)="D2":BS$(26)="A205EC#":BS$(27)="D<BAA":BS$(28)="A2"
200 AS$(29)="05L8DC#D4":AS$(30)="<F#EF#4AGA4":BS$(29)=AS$(29):BS$(30)=AS$(30)
210 AS$(31)="D205DEF#D":AS$(32)="E4E4EF#GE":BS$(31)="D2A405D4":BS$(32)="D4C#<B>C#DEC#>"
220 AS$(33)="F#4AD4C#D4":AS$(34)="<F#EF#4AGA4":BS$(33)="D404A4>D4C#D4":BS$(34)="04L8F#EF#4AGA4"
230 AS$(35)="04L8D2F#EDC#":AS$(36)="<B4B4B>C#D<B":BS$(35)="04D2R2":BS$(36)="L8BAGF#E4E4"
240 AS$(37)="05E4<E4>EDC#<B":AS$(38)="A4A4AB>C#<A":BS$(37)="EF#G#E4A4<A":BS$(38)=">AG#F#ED4D4"
250 AS$(39)="05D4<D4>D<C#D4":AS$(40)="<L8G#F#G#4BAB4":BS$(39)="04L8DEF#EDC#D4":BS$(40)=">GF#G#4BAB4"
260 AS$(41)="05G#2BAG#D":AS$(42)="C#<B>C#EDC#C#D#F#":BS$(41)="05D2L4<G#G#":BS$(42)="A4AA"
270 AS$(43)="05L8B4C#D4":AS$(44)="C#<B4>C#EDC#<B":AS$(45)="<A2":BS$(43)="04B8>C#AGL4C#E#":BS$(44)="<C#<AAG#8A8":BS$(45)="E2"
280 AS$(46)="04L8C#E4":AS$(47)="C#<B4>C#4E4D4":BS$(46)=AS$(46):BS$(47)=AS$(47)
290 AS$(48)="03A2>>L4GF#":AS$(49)="F#E2D":BS$(48)="03A2>>L4ED":BS$(49)="DC#2B"
300 AS$(50)="05L8C#4L8D<C#D#F#":AS$(51)="F#EF#BBGA4":BS$(50)="04B4A#4B2":BS$(51)="B2<C#2"
310 AS$(52)="05A2L8GF#EG":AS$(53)="F#4<B4>C#4A#4":BS$(52)="05D82L4E<A#":BS$(53)="BDEC#>"
320 AS$(54)="L8D82F#EF#G":AS$(55)="AGAB>C<B>C#D":BS$(54)="04L2DD":BS$(55)="D1"
330 AS$(56)="L805C#<B>C#C#<B>C#":AS$(57)="<B#B>AGF#ED":BS$(56)="04D2A4A4":BS$(57)="A4G8F#8E4A4"
340 AS$(58)="05C#4A4L8F#EDC#":AS$(59)="<B4B4B>C#D<B":BS$(58)="04L8AGF#ED4D4":BS$(59)="DEF#D4G4G"
350 AS$(60)="L805C#<B4A>D":AS$(61)="<A>D<F#E8F#8":AS$(62)="D2":BS$(60)="04G2L4F#A":BS$(61)="F#DDC#8D8":BS$(62)="<A2"
360 C$(1)="04L4DD":C$(2)="A2AA":C$(3)="A2AC#>"
370 C$(4)="04L4F#EF#E":C$(5)="D2DD"
380 C$(6)="04L4A2AA":C$(7)="A2BG#>"
390 C$(8)="04L4EEEE":C$(9)="ED<A#>"
400 C$(10)="04L4DDC#<B":C$(11)=">C#2"
410 C$(12)="04L4C#<A":C$(13)="AB8>C#8D<B"
420 C$(14)="03L4B2EE":C$(15)="DF#ED"
430 C$(16)="04L4C#<B#":C$(17)="BA#>>C#8F#F#>"
440 C$(18)="04L4F#GGF#8E8":C$(19)="DC#F#>"
450 C$(20)="04L4D2DG":C$(21)="ED#8E8F#D"
460 C$(22)="03B2L4AA":C$(23)=">A#EE"
470 C$(24)="04L4E2F#D":C$(25)="D2<B>E"
480 C$(26)="04L8E2E4A":C$(27)="L8AGF#GE4E4":C$(28)="F#2"
490 C$(29)="05L8DC#D4":C$(30)="<F#EF#4AGA4"
500 C$(31)="04L4D2F#D":C$(32)="GEA#>"
510 C$(33)="04L8D4F#4D<C#D4":C$(34)="<F#EF#4AGA4"
520 C$(35)="04L8DEF#EDC#<B#":C$(36)=">G#4>BAG#F#ED"
530 C$(37)="04L8C#DED<C#BAG#":C$(38)="F#4>AGF#EDC#>"
540 C$(39)="03L8B>C#D<B>>D<C#D4":C$(40)=">G#F#G#4BAB4"
550 C$(41)="04L4B2EE":C$(42)="EEF#F#>"
560 C$(43)="04L4G#8AB>C#<A":C$(44)="AEEC#8D8":C$(45)="C#2"
570 C$(46)="04L8AG#A4":C$(47)="<C#<B>C#4EDE4"
580 C$(48)="03L8A2>A#G#A4":C$(49)="BA#64EDE4"
590 C$(50)="04F#4<F#4>F#2":C$(51)="F#1"
600 C$(52)="04L4F#2E#C#":C$(53)="D<GEF#>"
610 C$(54)="03L2BA":C$(55)="AF#>"
620 C$(56)="03F#2L4>F#D":C$(57)="GEC#D"
630 C$(58)="03A4L8B>C#D<C#B#":C$(59)="L4G>DE<B"
640 C$(60)="04L8C#<BAGA4F#4":C$(61)="D4A4A4GA":C$(62)="F#2"
650 D$(1)="04D2"
660 D$(2)="04L4D<C#<B8>C#<A"
670 D$(3)="04L4D<D8E8F#G8E8"
680 D$(4)="03L4F#GA<A"

```

690 D\$(5)="03L4DF#8A8>D2"
 700 D\$(6)="05L4DC#8<B8>C#<A"
 710 D\$(7)="04L4DC#B#E"
 720 D\$(8)="04L4C#<B8A8G#E"
 730 D\$(9)="03L4ADC#F#"
 740 D\$(10)="03L4DE#E"
 750 D\$(11)="02A2"
 760 D\$(12)="02L4A>C#"
 770 D\$(13)="03L4DC#D"
 780 D\$(14)="03L4EDEF#"
 790 D\$(15)="03L4GF#G#"
 800 D\$(16)="03L4F#G#8A#8B#A"
 810 D\$(17)="03L4G#>C#<A#F#"
 820 D\$(18)="03L4BGED8C#8"
 830 D\$(19)="03L4DEF#<F#"
 840 D\$(20)="02L4B>A8F#8G2"
 850 D\$(21)="03L4GF#8E8D#<B"
 860 D\$(22)="03L4EF#8G8A#<#"
 870 D\$(23)="03L4D<BG#"
 880 D\$(24)="02L4AB8>C#8D8E8F#"
 890 D\$(25)="03L4F#E8D8E8F#8G"
 900 D\$(26)="03L4GEC#<A"
 910 D\$(27)="03L4DGA<A"
 920 D\$(28)="03D2"
 930 D\$(29)="04L8DC#D4"
 940 D\$(30)="03L8F#EF#4AGA4"
 950 D\$(31)="03D2R2"
 960 D\$(32)="04L8R2DC#D4"
 970 D\$(33)="03L8F#EF#4AGA4"
 980 D\$(34)="03D2"
 990 D\$(35)="03D2"
 1000 D\$(36)="04L8R2DC#D4"
 1010 D\$(37)="03L8G#F#G#4AB4"
 1020 D\$(38)="03E1"
 1030 D\$(39)="03L8E2AG#A4"
 1040 D\$(40)="03L8C#C#4EDE4"
 1050 D\$(41)="02A2"
 1060 D\$(42)="03L8AG#A4"
 1070 D\$(43)="03L8C#C#4EDE4"
 1080 D\$(44)="02A2R2"
 1090 D\$(45)="04R2BA#B4"
 1100 D\$(46)="03L8DC#D4F#EF#4"
 1110 D\$(47)="02B2R2"
 1120 D\$(48)="03L8R2DC#D4"
 1130 D\$(49)="02F#EF#4AGA4"
 1140 D\$(50)="02D2R2"
 1150 D\$(51)="03L8R2>DC#D4"
 1160 D\$(52)="02L8F#EF#4AGA4"
 1170 D\$(53)="02D2"
 1180 E\$(1)="04A4L8>A#F#":E\$(2)="G2<A4>GE":E\$(3)="F#2"
 1190 E\$(4)="R204A4L8>A#F#"
 1200 E\$(5)="05L8G2<A4>GE":E\$(6)="F#2R2"
 1210 E\$(7)="04R2L8F#4>F#D":E\$(8)="E2<F#4EF#"
 1220 E\$(9)="04L8R2G4>GE":E\$(10)="F#4EDEF#EF#"
 1230 E\$(11)="05E4<A4R2"
 1240 E\$(12)="04R2A4>L8GE":E\$(13)="F#EDEF4.D":E\$(14)="D2"
 1250 E\$(15)="05L4D8D":E\$(16)="<F#&F#A#A"
 1260 E\$(17)="04D2R2"
 1270 E\$(18)="05L4R2D2":E\$(19)="<F#&F#A#A"
 1280 E\$(20)="04D2R2"
 1290 E\$(21)="05L8R2D4.<A":E\$(22)=">D4.EDEF#"
 1300 E\$(23)="05D2R2"
 1310 E\$(24)="05L4R2E8E":E\$(25)="A#A#E#":E\$(26)="<A2"
 1320 E\$(27)="05L8E4.A":E\$(28)="E4.F#EF#EF#"
 1330 E\$(29)="05E2R2"
 1340 E\$(30)="05L8R2F#4.B":E\$(31)="F#4.GF#GF#G"
 1350 E\$(32)="05F#2R2"
 1360 E\$(33)="04L4R2A>D":E\$(34)="DF#F#A"
 1370 E\$(35)="05A2R2"
 1380 E\$(36)="05R2L4D8D":E\$(37)="<F#&F#A#A":E\$(38)="D2"
 1390 F\$(1)="04F#4>F#8D8"
 1400 F\$(2)="F#8E4.<A4>E8<A8"
 1410 F\$(3)="A2R2"
 1420 F\$(4)="R2F#4>F#8D8"
 1430 F\$(5)="E2<A4>E8<A8"
 1440 F\$(6)="A2R2"
 1450 F\$(7)="R2E4E4"
 1460 F\$(8)="D4R4B4R4"
 1470 F\$(9)="A2R2"
 1480 F\$(10)="R2E4E4"
 1490 F\$(11)="D4D4(A8F#8A4"
 1500 F\$(12)="F#2"
 1510 F\$(13)=E\$(29)
 1520 F\$(14)=E\$(30)
 1530 F\$(15)=E\$(31)
 1540 F\$(16)=E\$(29)
 1550 F\$(17)=E\$(34)
 1560 F\$(18)=E\$(35)
 1570 F\$(19)=E\$(43)
 1580 F\$(20)=E\$(44)
 1590 F\$(21)=E\$(45)
 1600 F\$(22)=E\$(54)
 1610 F\$(23)=E\$(55)


```

1620 F$(56)="05F#2R2"
1630 F$(60)=E$(60)
1640 F$(61)=E$(61)
1650 F$(62)=E$(62)
1660 OPEN "DATA.M-3" FOR OUTPUT AS #1
1670 FOR I=0 TO 11:GOSUB *PLAY:NEXT I
1680 FOR I=1 TO 28:GOSUB *PLAY:NEXT I
1690 FOR I=12 TO 45:GOSUB *PLAY:NEXT I
1700 FOR I=29 TO 62:GOSUB *PLAY:NEXT I
1710 FOR I=46 TO 62:GOSUB *PLAY:NEXT I
1720 FOR I=1 TO 28:GOSUB *PLAY:NEXT I
1730 CLOSE:END
1740 *PLAY
1750 WRITE #1,A$(1),B$(1),C$(1),D$(1),E$(1),F$(1):RETURN

```

リスト9

```

10 ' [BOURREE]
20 ' Programed by D.Kobayashi in 1985
30 ' save "MUSIC4.M88"
40 DIM A$(34),B$(34),C$(34),D$(34),E$(34),F$(34)
50 A$(0)="T180Q804V12":B$(0)="T180Q804V12":C$(0)="T180Q804V1802"
60 D$(0)="T180Q804V1802":E$(0)="T180Q803V89":F$(0)="T180Q802V8808"
70 A$(1)="R4"
80 A$(2)="04R4A4>D4R4"
90 A$(3)=A$(2)
100 A$(4)="04R4F#8E8F#4G#4"
110 A$(5)="05A8G#8AB8BA8R4"
120 A$(6)="05R4F#4B4R4"
130 A$(7)=A$(6)
140 A$(8)="05R4A4C#8BB8A8G#8"
150 A$(9)="05A2."
160 A$(10)="05R4"
170 A$(11)="05R4E4A4R4"
180 A$(12)=A$(11)
190 A$(13)="05R4D4G4F#8E8"
200 A$(14)="05F#8G8F#8E8D4R4"
210 A$(23)=A$(1)
220 A$(24)=A$(1)
230 A$(25)="05R4L8F#GABAB"
240 A$(26)="05L8GABAG4R4"
250 A$(31)="05R4F#4B4R4"
260 A$(32)="05R4B4>D4R4"
270 A$(33)="05R4L8AGF#GAG"
280 A$(34)="05F#2."
290 B$(1)="04r4"
300 B$(2)="04R4L4F#AR4"
310 B$(3)=B$(2)
320 B$(4)="04R4>D8C#8D4D4"
330 B$(5)="05D4A8BA8A4R4"
340 B$(6)="05R4D4F#AR4"
350 B$(7)="05R4E4E4R4"
360 B$(8)="05R4F#4F#4E4"
370 B$(9)="05E2."
380 B$(10)="R4"
390 B$(11)="04R4A4>E4R4"
400 B$(12)=B$(11)
410 B$(13)="05R4F#4E4E4"
420 B$(14)="05D4D8D8<A4R4"
430 B$(23)="05R4F#4D4R4"
440 B$(24)="05R4F#4A4R4"
450 B$(25)="05R4L8DEF#GF#E"
460 B$(26)="05D4R4R2"
470 B$(31)="05R4L4DF#R4"
480 B$(32)="05R4L4EBR4"
490 B$(33)="05R4F#8E8D4E4"
500 B$(34)="05D2."
510 C$(1)="04F#8G8"
520 C$(2)="04L4A>D<AG"
530 C$(3)="04L4F#>D<F#E"
540 C$(4)="04D4F#8E8F#4G#4"
550 C$(5)="04L8AG#ABAGF#E"
560 C$(6)="04L4F#>G<BA"
570 C$(7)="04L4C#>B<D"
580 C$(8)="05L8C#<B>C#D<C#<BAG#>"
590 C$(9)="04A2."
600 C$(10)="05C#8D8"
610 C$(11)="05L4EAED"
620 C$(12)="05L4C#AC#<B"
630 C$(13)="04L4A>DGF#8E8"
640 C$(14)="05L8F#GF#DEDF#D"
650 C$(15)="04B#>F#8G2F#4"
660 C$(16)="04G#>D#8E2F#4"
670 C$(17)=C$(15)
680 C$(18)="04E8>D#8E2<#AB8B"
690 C$(19)="05L4C#EGF#>"
700 C$(20)="04L4F#>C#E#D"
710 C$(21)="05B4F#8E8D4C#4"
720 C$(22)="04B2.F#8G8"
730 C$(23)="04L4A>D<F#E"
740 C$(24)="04L4D>A<C#<B"

```

```

750 C$(25)="05C4F#8G8A4C4"
760 C$(26)="04L8BAGAB>D<C#E"
770 C$(27)="04L8D<C#<B>C#D#>EG"
780 C$(28)="05L8F#DEDF#AGB"
790 C$(29)="05A4G8AB8B4D4"
800 C$(30)="05L8C#<B>BAG#>C#<A"
810 C$(31)="04F#8<C#8D2#4"
820 C$(32)="05B8C#8D2C#8<B8"
830 C$(33)="05C#8<A8>A4D4E8C#8"
840 C$(34)="05D2."
850 D$(1)="04L8D"
860 D$(2)="04L4F#AF#C#"
870 D$(3)="04L4D#D<A#"
880 D$(4)="04L4A>C#<AE"
890 D$(5)="04L4EC#<A>C#"
900 D$(6)="04D4B8A8B4D4"
910 D$(7)="04E4>E8D8E4<G#4"
920 D$(8)="04L4EAF#E"
930 D$(9)="04E2."
940 D$(10)="04L8AB"
950 D$(11)="05L4C#EC#<G#"
960 D$(12)="04L4A>C#<AE"
970 D$(13)="04L4F#ABA"
980 D$(14)="04A2A4A4"
990 D$(15)="04B8>D#8E2<B4"
1000 D$(16)="04B8F#8G2D#4"
1010 D$(17)="04E8>D#8E2D#4"
1020 D$(18)="05E4>A8B8BA4C#4"
1030 D$(19)="04F#4>C#4E4<F#8G8#8"
1040 D$(20)="04L4AF#>G<B"
1050 D$(21)="04B4>D<C#<B4A#4"
1060 D$(22)="04F#2.D8E8"
1070 D$(23)="04L4F#DAB"
1080 D$(24)="05L4C<DAG"
1090 D$(25)="04F#4<C#<B8A4F#4"
1100 D$(26)="04L4GDG"
1110 D$(27)="04L4F#D>D<A"
1120 D$(28)="04L4AF#A>C#"
1130 D$(29)="05D8E8D<C#8<B4G#4"
1140 D$(30)="04L4AEF#G"
1150 D$(31)="04A8E8F#2B8AB"
1160 D$(32)="04G#8AB8B2G#4"
1170 D$(33)="04L4AAAA"
1180 D$(34)="04A2."
1190 E$(1)="R4"
1200 E$(2)="04R4F#8E8F#4R4"
1210 E$(3)="04R4AB8A#A4"
1220 E$(4)="04R4D8E8D4D4"
1230 E$(5)="04L4C#ED<A"
1240 E$(6)="03A4>F#8E8F#4R4"
1250 E$(7)="04R4B8AB4<B4"
1260 E$(8)="03L4E>C#D<B"
1270 E$(9)="04C#2."
1280 E$(10)="R4"
1290 E$(11)="05R4C#8<B8>C#4R4"
1300 E$(12)="04R4E8D8E4R4"
1310 E$(13)="04L4R4D<C#"
1320 E$(14)="04D4D8E8F#4R4"
1330 E$(15)="04R4E8D#8E4R4"
1340 E$(16)=E$(15)
1350 E$(17)="04R4E8D#8E4F#4"
1360 E$(18)="03B4>E8D8E4"
1370 E$(19)="04R4C#<B8>C#4R4"
1380 E$(20)="04R4A#8G8A#4B4"
1390 E$(21)="04L4F#BF#E"
1400 E$(22)="04D4D8C#8D4R4"
1410 E$(23)="03R4A8B8<C4R4"
1420 E$(24)="04R4F#8G8A4R4"
1430 E$(25)="04R4A4D"
1440 E$(26)="04L4D<B>E"
1450 E$(27)="04L4DF#F#A"
1460 E$(28)="04L4D<A>D<G"
1470 E$(29)="04L4F#DDE"
1480 E$(30)="04L4E<A>DE"
1490 E$(31)="04F#4D8C#8D4R4"
1500 E$(32)="04R4E8D8E4E4"
1510 E$(33)="04E4D8E8F#4E4"
1520 E$(34)="04F#2."
1530 F$(1)="R4"
1540 F$(2)="03R4D8C#8D4R4"
1550 F$(3)="03R4C#8<B8>C#4R4"
1560 F$(4)="02R4B8<C#8D4<B4"
1570 F$(5)="03L4C#<AF#A"
1580 F$(6)="02D4>D8C#8D4R4"
1590 F$(7)="02R4G8F#8G8W4E4"
1600 F$(8)="02L4AF#DE"
1610 F$(9)="01A2."
1620 F$(10)="02R4"
1630 F$(11)="02R4A8G#8A4R4"
1640 F$(12)="02R4G8F#8G4R4"
1650 F$(13)="02L4R4F#EA"
1660 F$(14)="02D4B8AB8D4D"
1670 F$(15)="02L4GBED"
1680 F$(16)="02L4C#<G#F#"

```

```

1690 F$(17)="02L4E>C<BA"
1700 F$(18)="02G4C#<B8>C#4R4"
1710 F$(19)="02R4A#<8G8A#<4R4"
1720 F$(20)="02R4F#<8E8F#4G4"
1730 F$(21)="02D8C#<8D8E#4F#4"
1740 F$(22)="01B4>B8A8B4R4"
1750 F$(23)="02R4F#<8G8A4R4"
1760 F$(24)="02R4D8E8F#4R4"
1770 F$(25)="02R4A8G8F#4D4"
1780 F$(26)="01G4>G8F#8G4<A4"
1790 F$(27)="01B4>B8A8B4C4"
1800 F$(28)="02D4>D8C#8D4E4"
1810 F$(29)="02F4B4A8G8F#8E8"
1820 F$(30)="02L4AGF#E"
1830 F$(31)="02D4B8A8B4R4"
1840 F$(32)="02R4G8F#8G#4E4"
1850 F$(33)="02A8G8F#8G8A4<A4"
1860 F$(34)="02D2."
1870 OPEN "DATA.M-4" FOR OUTPUT AS #1
1880 FOR I=0 TO 9:GOSUB *PLAY:NEXT I
1890 FOR I=1 TO 34:GOSUB *PLAY:NEXT I
1900 FOR I=10 TO 34:GOSUB *PLAY:NEXT I
1910 CLOSE:END
1920 *PLAY
1930 WRITE #1,C$(1),D$(1),E$(1),F$(1),A$(1),B$(1):RETURN

```

●● リスト10

```

10 ' [ GIGUE ]
20 ' Programed by D.Kobayashi in 1985
30 ' save "MUSIC5.M88"
40 DIM A$(74),B$(74),C$(74),D$(74),E$(74),F$(74)
50 A$(0)="T15004V8Q8802":B$(0)="T15004V8Q882":C$(0)="T15004V8Q88"
60 D$(0)="T15003V8Q8800":E$(0)="T15004V11Q8":F$(0)="T15004V11Q8"
70 A$(1)="04D8"
80 A$(2)="04L4A.B."
90 A$(3)="04L8A4.&AB>C#"
100 A$(4)="05L8DC#<BAG#"
110 A$(5)="04L8F#E#DF#A#"
120 A$(6)="05L4D.F.#"
130 A$(7)="05L4E.&ED8"
140 A$(8)="05L8C#<BAG>C#D"
150 A$(9)="05L8C#<B>C#<A>C#E"
160 A$(10)="05L8AGAAGA"
170 A$(11)="05L8AGA<B>D#F#"
180 A$(12)="05L8GF#GGF#G#"
190 A$(13)="05L8GF#G<A>C#E"
200 A$(14)="05L8F#GAAGA"
210 A$(15)="05L8DEF#F#E#"
220 A$(16)="04L8B>C#D<D<D"
230 A$(17)="04L8GA8E8G#B"
240 A$(18)="05L8EDC#D<C#B"
250 A$(19)="04L8A4>F#<A4>F#"
260 A$(20)="04L8G#4>C<G4>E"
270 A$(21)="04L8F#4D<F4D"
280 A$(22)="04L8E4.&EG#B"
290 A$(23)="05L8DC#D<C#B"
300 A$(24)="05L8C#1D16<CAEAG#"
310 A$(25)="04L4A.&A"
320 A$(26)="04A8"
330 A$(27)="05L4E.F.#"
340 A$(28)="05L8E4.&E#F#G#"
350 A$(29)="05L8F#EDC#D<C#B"
360 A$(30)="04L8ABW>C#<F#A#>C#"
370 A$(31)="05L4F#.#"
380 A$(32)="05L8G32F#4&F#16.F#ED"
390 A$(33)="05L8C#D<BF#BA#"
400 A$(34)="04L8B4.&B>D#C#"
410 A$(35)="05L8DC#D<D<D"
420 A$(36)="05L8DC#D<C#G#B"
430 A$(37)="05L8EDC#B>C#D"
440 A$(38)="05L8C#<B>C#<A>C#E"
450 A$(39)="05L8GF#GGF#G#"
460 A$(40)="05L8GF#G<A>C#E"
470 A$(41)="05L8AGF#F#F#G#"
480 A$(42)="05L8F#E#F#DF#E#"
490 A$(43)="05L8F#G#AGAB"
500 A$(44)="05L8C#4.&C#GB"
510 A$(45)="05L8AGF#F#EDC#"
520 A$(46)="04L8B4.&B#F#A"
530 A$(47)="05L8G#F#E#F#ED"
540 A$(48)="05L8C#F#EDC#<B>"
550 A$(49)="04L8AGF#F#AG#"
560 A$(50)="04L4F#.#&F#D8"
570 A$(51)="04L4A.B."
580 A$(52)="05L8C4.&C#BA"
590 A$(53)="04L8B>D<A>G#F#"
600 A$(54)="05L8GBA4.&"
610 A$(55)="05L8BA8F#GE"
620 A$(56)="05L8F#AGA4.&"
630 A$(57)="05L8AGF#F#E#D#"
640 A$(58)="05L8EGF#G4.&"
650 A$(59)="05L8GF#EDC#"
660 A$(60)="05L8DF#E#F#GA"
670 A$(61)="04L8B>C#D#F#"
680 A$(62)="05L8C#E#F#GE"
690 A$(63)="05L8DF#GAF#"
700 A$(64)="05L8E#F#GAB#"
710 A$(65)="05L8F#4.&F#ED#"
720 A$(66)="05L8EC#D<A>DC#"
730 A$(67)="04L4D>D8A.&"
740 A$(68)="05L8ABA8AGF#"
750 A$(69)="05L8AGAF#E#"
760 A$(70)="05L8GF#E#F#GE"
770 A$(71)="05L8C#D<A4.&"
780 A$(72)="04L4A>A8B>C#B"
790 A$(73)="06L8D4<AF#ED"
800 A$(74)="05L4D.&D"
810 B$(1)="04D8"
820 B$(2)="04L4F#.#"
830 B$(3)="04L4B.&F#GB"
840 B$(4)="04L4AGF#E#"
850 B$(5)="04L8D<D<A>D#E"
860 B$(6)="04L4F#.#"
870 B$(7)="04L4B.&B8B"
880 B$(8)="04L4A8A8G"
890 B$(9)="04L8AG#A8E#"
900 B$(10)="05L4C#R8C#R8"
910 B$(11)="04L4B.&B8B"
920 B$(12)="04L4BR8BR8"
930 B$(13)="04L4A.R4>C#B"
940 B$(14)="05L4D#F#E<A8"
950 B$(15)="04L4B>D8D8"
960 B$(16)="05L4D<B8A>D8"
970 B$(17)="05L4E<E8G#G#B"
980 B$(18)="04L4B.&B8B"
990 B$(19)="04F#2."
1000 B$(20)="04E2."
1010 B$(21)="04D2.&"
1020 B$(22)="04L8D4.&DEG#"
1030 B$(23)="04L4BG#B8B8"
1040 B$(24)="04L4AE8E8B"
1050 B$(25)="04E4.&E4"
1060 B$(26)="05C#B"
1070 B$(27)="05C#4.D4."
1080 B$(28)="05C#4.&C#B8DEB"
1090 B$(29)="05D4<F#B8E4B"
1100 B$(30)="04L8F#4.&F#>F#E"
1110 B$(31)="05L4D.E."
1120 B$(32)="05L4D.&D<F#B"
1130 B$(33)="04L4EE8F#F#B"
1140 B$(34)="04L4F#.#F#F#"
1150 B$(35)="04L4BR8BR8"
1160 B$(36)="04L4GR8BR8"
1170 B$(37)="04L4BA8AG#B"
1180 B$(38)="04L8AG#AEA>C#"
1190 B$(39)="05L4ER8C#R8"
1200 B$(40)="05L4C#R8EE8"
1210 B$(41)="05L4ED8DC8"
1220 B$(42)="05L8D<D<A>DC#"
1230 B$(43)="05L8D4.&D<D"
1240 B$(44)="04L4E.#E#>C#B"
1250 B$(45)="05L8C#4.&C#<BA"
1260 B$(46)="04L4B.&BA"
1270 B$(47)="05L4DC#<B8B"
1280 B$(48)="04L8A4>C#<BAG#"
1290 B$(49)="04L4F#F#E#B#"
1300 B$(50)="04L4C.&C#D#"
1310 B$(51)="04L4F#.#"
1320 B$(52)="04L8F#4.&F#GA"
1330 B$(53)="04L4GB8A8"
1340 B$(54)="04L8B>D<D4<B"
1350 B$(55)="05L4EDC#B.&"
1360 B$(56)="05L8C#F#ED#4E"
1370 B$(57)="05L4D#F#B<B.&"
1380 B$(58)="04L8B4.&ED"
1390 B$(59)="05L4C#E#<A.&"
1400 B$(60)="04L4A8A8"
1410 B$(61)="04L4GR8BR8"
1420 B$(62)="04L4AR8>C#R8"
1430 B$(63)="04L4BR8>DR8"
1440 B$(64)="05L4ER8ER8"
1450 B$(65)="05L8D<D<B4B"
1460 B$(66)="05L4C#<A8BA"
1470 B$(67)="04F#4.R4>C#B"
1480 B$(68)="04B2."
1490 B$(69)="04A2."
1500 B$(70)="04G2.&"
1510 B$(71)="04L8G4.&GF#>C#"
1520 B$(72)="05L4EC#B8B"
1530 B$(73)="05L8D4F#D<D"
1540 B$(74)="04L4A.&A"
1550 C$(1)="04D8"
1560 C$(2)="04L4D.D."
1570 C$(3)="04L4D.&D<G#"

```

1580 C\$(4) = "04L4DD8DC#8"
 1590 C\$(5) = "03L8AGAF#A>C#"
 1600 C\$(6) = "04L4D. D."
 1610 C\$(7) = "04L4E. 8EE8"
 1620 C\$(8) = "04L8E4C#F#4E"
 1630 C\$(9) = "04L8EDEC#EG"
 1640 C\$(10) = "04L4C#R8F#R8"
 1650 C\$(11) = "04F#4. R4D#8"
 1660 C\$(12) = "04L4ER8ER8"
 1670 C\$(13) = "04E4. R4A8"
 1680 C\$(14) = "04L4A4D8E4E8"
 1690 C\$(15) = "04L4F#C#B8B8"
 1700 C\$(16) = "04L4E8F#F#8"
 1710 C\$(17) = "03L4B>G#8BD8"
 1720 C\$(18) = "04L4C#E8ED8"
 1730 C\$(19) = "04C#4. <B4. &"
 1740 C\$(20) = "03B4. A4. &"
 1750 C\$(21) = "03A4. L8G#8A"
 1760 C\$(22) = "03L8B>C#D>G#B>D"
 1770 C\$(23) = "04L4G#E8C#A>E8"
 1780 C\$(24) = "04L4E8C#B8B8"
 1790 C\$(25) = "04L4C#.. 8C#"
 1800 C\$(26) = "04E8"
 1810 C\$(27) = "04L4A. A. "
 1820 C\$(28) = "04A. 8AC#8"
 1830 C\$(29) = "03L4AB8B8"
 1840 C\$(30) = "04L8C#4A#B>C#<A#"
 1850 C\$(31) = "03L4B. >B."
 1860 C\$(32) = "04L4B. 8B<B8"
 1870 C\$(33) = "03L4GB8>C#C#8"
 1880 C\$(34) = "04L4D. 8DF#8"
 1890 C\$(35) = "04L4ER8ER8"
 1900 C\$(36) = "03L4BR8>EE8"
 1910 C\$(37) = "04L4EF#8F#E8"
 1920 C\$(38) = "04L8EDEC#E8A"
 1930 C\$(39) = "04L4AB8A8"
 1940 C\$(40) = "04L8E8A8"
 1950 C\$(41) = "04L4AB8BA8"
 1960 C\$(42) = "04L8AGAF#DA"
 1970 C\$(43) = "04L8AGF#B4A"
 1980 C\$(44) = "04L4G#.. 8G#E8"
 1990 C\$(45) = "04L8F#G#A<A4>C#"
 2000 C\$(46) = "04L4F#.. 8F#>C#8"
 2010 C\$(47) = "04L8BAGF#E#8"
 2020 C\$(48) = "04L4F#.. 8F#G#8"
 2030 C\$(49) = "04L4C#D8<G#>C#8"
 2040 C\$(50) = "03L4A. 8AA8"
 2050 C\$(51) = "04L4D. D. "
 2060 C\$(52) = "03L4A. 8A>D8"
 2070 C\$(53) = "04L4DE8E4D8"
 2080 C\$(54) = "04L4D. 8D8GBF#8"
 2090 C\$(55) = "04L4B8B8"
 2100 C\$(56) = "04L4F#>C#<A#E8"
 2110 C\$(57) = "04L4F#A8GF#8"
 2120 C\$(58) = "04L4E<B8>B4F#8"
 2130 C\$(59) = "04L4EG8F#8"
 2140 C\$(60) = "04L4D<A8>DD8"
 2150 C\$(61) = "04L4DR8ER8"
 2160 C\$(62) = "04L4ER8F#8"
 2170 C\$(63) = "04L4FR8DR8"
 2180 C\$(64) = "04L4AR8A8"
 2190 C\$(65) = "04L4AB8B8AGF#8"
 2200 C\$(66) = "04L4EF#8EE8"
 2210 C\$(67) = "04L4D. R4F#8"
 2220 C\$(68) = "04L4F#.. E. &"
 2230 C\$(69) = "04L4E. D. &"
 2240 C\$(70) = "04L4D. C#8E8D8"
 2250 C\$(71) = "04L8EF#>C#E8G"
 2260 C\$(72) = "05L4C#<A8D8G"
 2270 C\$(73) = "04L4A>D8<A8GBF#8"
 2280 C\$(74) = "04L4F#.. 8F#"
 2290 D\$(1) = "02D8"
 2300 D\$(2) = "03L8DCCDDC#D"
 2310 D\$(3) = "03L8DCD#D4E"
 2320 D\$(4) = "03L8F#>A4A4"&br/>
 2330 D\$(5) = "03L8D4. 8D4C#"
 2340 D\$(6) = "02L8BABAG#A"
 2350 D\$(7) = "02L8G#F#G#EF#G#"
 2360 D\$(8) = "02L8A4>F#D4E"
 2370 D\$(9) = "02L8A4. 8A4G"
 2380 D\$(10) = "02L8F#4>GF#4E"
 2390 D\$(11) = "03L8D#4F#D#4<B"
 2400 D\$(12) = "03L8E4F#E4D"
 2410 D\$(13) = "03L8C#4EC#4A"
 2420 D\$(14) = "03L8D4>D#C4A#C#"
 2430 D\$(15) = "02L8B4>BA4A#A"
 2440 D\$(16) = "02L8B4>G#F#4C#F#"
 2450 D\$(17) = "02L8E4>EDC#<B#"
 2460 D\$(18) = "03L8C#<BA>E4. &"
 2470 D\$(19) = "03L8EF#ED#4>C#"
 2480 D\$(20) = "03L8EDC#<A#>B"
 2490 D\$(21) = "03L8DCC#BG#A"
 2500 D\$(22) = "02L8G#ABE4. &"
 2510 D\$(23) = "02L8E4>EF#4G#"
 2520 D\$(24) = "03L8A4C#E4C#E"
 2530 D\$(25) = "02A48A4. &"
 2540 D\$(26) = "02AB8"
 2550 D\$(27) = "03L8AG#AAG#A"
 2560 D\$(28) = "03L8AG#AC#4A#"
 2570 D\$(29) = "03L8D4E4G"
 2580 D\$(30) = "03L8F#4#E4D4C#"
 2590 D\$(31) = "02L8BA#B8B8B#"
 2600 D\$(32) = "02L8B>C#DDC#<B#"
 2610 D\$(33) = "03L8E#F#G#EF#F#"
 2620 D\$(34) = "02L8B>D#F#B4A"
 2630 D\$(35) = "03L8B#4B84C#"
 2640 D\$(36) = "03L8E4C#G#E4E"
 2650 D\$(37) = "03L8C#4F#D4E4"
 2660 D\$(38) = "02L8A4>AA4E"
 2670 D\$(39) = "03L8C#4EE4C#"
 2680 D\$(40) = "02L8A4>C#B84A#A"
 2690 D\$(41) = "02L8F#4B8G4A"
 2700 D\$(42) = "02L8D4>D#F#4A#"
 2710 D\$(43) = "03L8B4C#<B4. &"
 2720 D\$(44) = "03L8G#E#C#E#C#<B#"
 2730 D\$(45) = "03L8F#4<F#>F#4E"
 2740 D\$(46) = "03L8DD#F#D#F#"
 2750 D\$(47) = "03L8B4>C#D4<G#"
 2760 D\$(48) = "03L8A4F#D4E#"
 2770 D\$(49) = "03L8F#4C#4C#<B#"
 2780 D\$(50) = "02L8F#A>C#F#EF#F#"
 2790 D\$(51) = "03L8DCCDDC#D"
 2800 D\$(52) = "03L8D#F#4A>D4C#F#"
 2810 D\$(53) = "03L8G4EC4D"
 2820 D\$(54) = "02L8G4. 8GB>D"
 2830 D\$(55) = "03L8G4E4A4. &"
 2840 D\$(56) = "03L8A4<E#F#A>C#"
 2850 D\$(57) = "03L8F#4#D#G4. &"
 2860 D\$(58) = "03L8G4<D#EGB#"
 2870 D\$(59) = "03L8E4C#D#4. &"
 2880 D\$(60) = "03L8F#4<C#D4F#"
 2890 D\$(61) = "02L8G4B8G#4E"
 2900 D\$(62) = "02L8A4>C#A#4F#"
 2910 D\$(63) = "02L8B4B4B#"
 2920 D\$(64) = "03L8C#4C#>C#4C#A#"
 2930 D\$(65) = "03L8D#E#G4. &"
 2940 D\$(66) = "03L8G4F#G4A"
 2950 D\$(67) = "03L8BAGF#E#8"
 2960 D\$(68) = "03L8D#4B8D#"
 2970 D\$(69) = "03L8C#4AC4A#C#"
 2980 D\$(70) = "02L8B4>G<B-4>G#"
 2990 D\$(71) = "02L8A4. 8A>C#E#"
 3000 D\$(72) = "03L8G>C#E#<G#E#"
 3010 D\$(73) = "03L8F#EDA4C#"
 3020 D\$(74) = "03D4. 8D4"
 3030 E\$(1) = "05D8"
 3040 E\$(2) = "05L4A. B. "
 3050 E\$(3) = "05L8A4. 8AB8C#"
 3060 E\$(4) = "06L8DCC#<BAG#"
 3070 E\$(5) = "05L8F#EF#D4R8"
 3080 E\$(9) = "05R4RR8E8"
 3090 E\$(10) = "05L8AGAG#A"
 3100 E\$(11) = "05L8A4. 8ABA"
 3110 E\$(12) = "05L8G#F#G#G#F#"
 3120 E\$(13) = "05L8C4. 8GAC#"
 3130 E\$(14) = "05F#8RR8RR4R8"
 3140 E\$(23) = "06R4RR84R8"
 3150 E\$(24) = "06L8C#<BABA#G#"
 3160 E\$(25) = "05A4. 8A4"
 3170 E\$(26) = "04AB8"
 3180 E\$(27) = "05L4E. F#.. "
 3190 E\$(28) = "05L4E. 8ER8"
 3200 E\$(30) = "04R4RR84F#8"
 3210 E\$(31) = "05L4F#.. G. "
 3220 E\$(32) = "05L4F#.. 8F#R8"
 3230 E\$(34) = "04R4RR84F#8"
 3240 E\$(35) = "05L8DEDED#"
 3250 E\$(36) = "05L8D4. 8DED#"
 3260 E\$(37) = "05L4E8ABG#8"
 3270 E\$(38) = "05L4E8ABE#A#"
 3280 E\$(39) = E\$(12)
 3290 E\$(40) = E\$(13)
 3300 E\$(41) = "05L8AG#EF#F#G#"
 3310 E\$(42) = "05L8F#EF#D4R8"
 3320 E\$(50) = "05R4RR84R8"
 3330 E\$(51) = "05L4A. B. "
 3340 E\$(52) = "06L8C4. 8C#BA"
 3350 E\$(53) = "05L8BAG#F#"
 3360 E\$(54) = "05G4RR84R8"
 3370 E\$(65) = "05L8R4A8C#D#"
 3380 E\$(66) = "05L8ABAG#F#E#"
 3390 E\$(67) = "05D4RR84R8"
 3400 E\$(72) = "06R4RR84R8#8"
 3410 E\$(73) = "06L8D4<A#F#ED#"
 3420 E\$(74) = "05L4D. 8D"
 3430 F\$(1) = "04AB8"
 3440 F\$(2) = "05L4D. G. "
 3450 F\$(3) = "05L4D. 8DGB#"


```

3460 F$ (4) = "05L4AG8F#E#"
3470 F$ (5) = "05L4D<A8AR#"
3480 F$ (9) = "05R4R8R4E#"
3490 F$ (10) = "05L8F#EF#F#E#"
3500 F$ (11) = "05L8F#4.8F#GF#"
3510 F$ (12) = "05L8DEDE#"
3520 F$ (13) = "05L8E4.8F#E#"
3530 F$ (14) = "05D8R8R8R4E#"
3540 F$ (23) = "05R4R8R4B#"
3550 F$ (24) = "05L4AE8EE#"
3560 F$ (25) = "05L4E.8E#"
3570 F$ (26) = "04AB#"
3580 F$ (27) = "041BaF#a>d<a>d"
3590 F$ (28) = "04L4A.8AR#"
3600 F$ (29) = "05R4R8R4E#"
3610 F$ (31) = "05L4D.E."
3620 F$ (32) = "05L4D.8DR#"
3630 F$ (36) = "05R4R8R4G8#"
3640 F$ (37) = "05L4AF#8F#E#"
3650 F$ (38) = "05L4E<A8AR#"
3660 F$ (40) = "R4R8R4E#"
3670 F$ (41) = "05L4ED8D<A#"
3680 F$ (42) = "04L4A>D8<AR#"
3690 F$ (50) = "R4R8R4O4A#"
3700 F$ (51) = "05L4D.G."
3710 F$ (52) = "05L4F#.8F#F#8#"
3720 F$ (53) = "05G4R8R4B#"
3730 F$ (54) = "05D4R8R4B#"
3740 F$ (65) = "05L4E4F#DEF#"
3750 F$ (66) = "05L8E4F#ED<A#"
3760 F$ (67) = "04F#4R8R4R#"
3770 F$ (72) = "05R4R8R4E#"
3780 F$ (73) = "05L8AGF#F#GA#"
3790 F$ (74) = "05L4F#.8F#"
3800 OPEN "DATA.M5" FOR OUTPUT AS #1
3810 FOR I=0 TO 25:GOSUB *PLAY:NEXT I
3820 FOR I=1 TO 74:GOSUB *PLAY:NEXT I
3830 FOR I=26 TO 74:GOSUB *PLAY:NEXT I
3840 CLOSE:END
3850 *PLAY
3860 WRITE #1,A$(1),B$(1),C$(1),D$(1),E$(1),F$(1)
3870 RETURN

```

リスト11

```

10 * [Brandenburgische Konzert Nr.6 B-Dur BWV.1051]
20 * Programmed by D.Kobayashi in 1985
30 * save "MUSIC6.END"
40 DIM A$(61),B$(61),C$(61),D$(61),E$(61),F$(61)
50 A$(0) = "0477V9Q80#"
60 A$(1) = "R1." : A$(2) = A$(1) : A$(3) = A$(1) : A$(4) = A$(1)
70 A$(5) = "05R2P262"
80 A$(6) = "04A1B-2"
90 A$(7) = "04L4R4B->CE-8D8E-E-"
100 A$(8) = "05L4E-D8C8D8E-8F#"
110 A$(9) = "05L8F4E-DE-4FG4F#"
120 A$(10) = "05F4.G8D2C4.CB-8"
130 A$(11) = "04L8B-F>D<CB-A-GFE-DE-4"
140 A$(12) = "04L8E-G>C<B-A-GFE-DCD4"
150 A$(13) = "04L8DFB-A-GFE-F16G16CE-B-A"
160 A$(14) = "04B-<B>4->B>2>C2"
170 A$(15) = "04D1E-2"
180 A$(16) = "04R4L4E-FA-8GBA-A-"
190 A$(17) = "04L8A-4GF4GA-B-A-4B-4"
200 A$(18) = "04L8B-4A-4<C<B->C4<B-A-"
210 A$(19) = "04B-4.>C8<G2F4.E-8"
220 A$(20) = "04L2E-GA-"
230 A$(21) = "03B1O4C2"
240 A$(22) = "04L4R4CDF8E-8F#"
250 A$(23) = "04L8F4EDEFF4F.G"
260 A$(24) = "04L8GB-A-GFE-D-CC<B-A-4"
270 A$(25) = "03L8B->D-GFEDCC<B-AB-A"
280 A$(26) = "03L8A->CFE-D-CC<B->C16D-16<GB->FE"
290 A$(27) = "04F1." : A$(28) = A$(27)
300 A$(29) = "04L8FA-GFF2E4.F8"
310 A$(30) = "04F2R2R2"
320 A$(31) = A$(1) : A$(32) = A$(1) : A$(33) = A$(1)
330 A$(34) = "05R2E-2F2"
340 A$(35) = "04G1A-2"
350 A$(36) = "04L4R4A-B->D-8C8D-D-"
360 A$(37) = "05L4D-C8<B-8C8D-8E-E-"
370 A$(38) = "05L8E-4D-CC-4FE-F4E-D-"
380 A$(39) = "05E-4.F8C2<B4.A-8"
390 A$(40) = "04L8A-ED->C<CB-A-GFE-DE-DE-"
400 A$(41) = "04L4FGB-8A-8B-B-"
410 A$(42) = "04L8B-CA-GFE-DC<B>C<B>C"
420 A$(43) = "04L8D4.A-GFE-DC<B>CE-16D16"
430 A$(44) = "04L8E-G>E-DC<B-AGF#GF#"
440 A$(45) = "04L8A4.>E-DC<B-AGF#GB-16A16"

```

```

450 A$(46) = "04L8B-4>E-4E-FE-DC<B-AB-16>C16"
460 A$(47) = "04L8F#4D4G4.DC<B-AB-16G16"
470 A$(48) = "04D2>D2E-2"
480 A$(49) = "04F#1G2"
490 A$(50) = "04R4L4GA->C<B-8>CC"
500 A$(51) = "05L8C4<B-AB-4>DCD4D4"
510 A$(52) = "05L8D4C<B->C4E-DE-4DC"
520 A$(53) = "05L8D4.E-<B-2AGA4"
530 A$(54) = "V804G1."
540 A$(55) = "05C1."
550 A$(56) = "04F#1."
560 A$(57) = "04L4F#DV18E8F#8GG"
570 A$(58) = "04L8GB-AGF#2.C4"
580 A$(59) = "04L4GV88B->CE-8D8E-2"
590 A$(60) = "05L4E-<AB>D8C8D2"
600 A$(61) = "05L8DE-DC<B-AGF#GL16AB-AB-8>C<F#1."
610 B$(0) = "T7004V9Q80#"
620 B$(1) = "04R2B-2>C2"
630 B$(2) = "04D1E-2"
640 B$(3) = "04R4L4E-FA-8GBA-A-"
650 B$(4) = "04L8A-4GF4GA-BA-B-"
660 B$(5) = A$(1) : B$(6) = A$(12)
670 B$(7) = "04L8DFB-AGFE-F16G16CE-B-A"
680 B$(8) = "04B-1."
690 B$(9) = B$(8)
700 B$(10) = "04L8B->DC<B-B-2A4.B-"
710 B$(11) = "04L2B-FG"
720 B$(12) = "03A1B-2"
730 B$(13) = "03L4R4B->CE-8D8E-E-"
740 B$(14) = "04L8E-DCDE-D-CC<B-A-GA-4"
750 B$(15) = "03L8A->CFE-DC<B-A-GF4"
760 B$(16) = "03L8GB>E-DC<B-A-B-16>C16<FA>E-D"
770 B$(17) = "05E-1."
780 B$(18) = B$(17)
790 B$(19) = "05L8E-GFE-E-2D4.E-"
800 B$(20) = "05E-1R2"
810 B$(21) = "R1."
820 B$(22) = B$(21)
830 B$(23) = B$(21)
840 B$(24) = "05R2C2D-2"
850 B$(25) = "04E1F2"
860 B$(26) = "04L4R4FGB-8A-8B-B-"
870 B$(27) = "04L4B-A-8GBA->C8<B-8>CC"
880 B$(28) = "05L8C4<B-A-B-4>D-CD-4C<B-"
890 B$(29) = "05L4C.D-8<A-2G.F8"
900 B$(30) = "04L8F2A-2>C<B->C4"
910 B$(31) = "04D1E-2"
920 B$(32) = "04R4L4E-FA-8GBA-A-"
930 B$(33) = "04L8A-4GF4GA-BA-A-4.G16A-16"
940 B$(34) = "04L8B->D-CC<B-A-GFE-D-CD-4"
950 B$(35) = "04L8D-FB-A-GFE-D-CC<B->C4"
960 B$(36) = "04L8CE-A-GFE-D-E-16F16<B->D-A-G"
970 B$(37) = "04A-1."
980 B$(38) = B$(37)
990 B$(39) = "04L8A->C<B-A-A-2G4.A-"
1000 B$(40) = "04L4E-E-FA-8GBA-A-"
1010 B$(41) = "04L8A->C<B-A-GFEG16F16GB-A-G"
1020 B$(42) = "05L4C<CDF9E-8F#"
1030 B$(43) = "04L8FA-GFE-DCE-16D16E-DE-4"
1040 B$(44) = "04L4E-GA>C8<B-8>CC"
1050 B$(45) = "05L8CE-DC<B-AGB-16A16B-AB-4"
1060 B$(46) = "04L8B->C<B->CE-16D16E-DCD16E-16"
1070 B$(47) = "04L8A4>D4C<B-AGFE-DG16F#16G"
1080 B$(48) = "04L8F#DEF#GFE-DC<B>C4"
1090 B$(49) = "04L8CE-AGF#EDCC<B-AB-4"
1100 B$(50) = "03L8B->DGE-DCD16E-16<A>CGF#"
1110 B$(51) = "04G1."
1120 B$(52) = B$(51)
1130 B$(53) = "04L8GB-AGG2F#4.G"
1140 B$(54) = "V804F1."
1150 B$(55) = "04E1."
1160 B$(56) = "04E-1."
1170 B$(57) = "04L4E-V10<F#GB-8A8B->D8C8"
1180 B$(58) = "04L4D.E-8<B-2A4.G8"
1190 B$(59) = "03G2R4V88L8GFE-DCD16E-16"
1200 B$(60) = "03A2R4L8FPE-DC<B->C16D16"
1210 B$(61) = "03G4.>L8E-DC<B-AB-G>DCD1"
1220 C$(0) = "T7008L4V86"
1230 C$(1) = "03L4E-FGE-A-F"
1240 C$(2) = "03L4B-A-B-G<C<B-"
1250 C$(3) = "03L4A-GA-FB-C"
1260 C$(4) = "03L4E-FE-DECE"
1270 C$(5) = "03L4DCD<B>E-C"
1280 C$(6) = "03L4FE-FDGE"
1290 C$(7) = "03L4E-DE-CC<F"
1300 C$(8) = "02L4B->FB>C<B-A-"
1310 C$(9) = "03L4GA-GFE-G"
1320 C$(10) = "03L4DE-FE-F<F"
1330 C$(11) = "02L4B->CD<B>E-C"
1340 C$(12) = "03L4FE-PDFG"
1350 C$(13) = "03L4E-DE-CC<F"
1360 C$(14) = "02L4B-A-GF-A-F"
1370 C$(15) = "02L4B-A-B-G<C<B-"
1380 C$(16) = "02L4A-GA-FB-B-"

```

1390 C\$(17)="02L4E->B->E-FE-D-"
 1400 C\$(18)="04L4CD-<C(B-A)->C"
 1410 C\$(19)="03L4GA-B-A-B-<B-"
 1420 C\$(20)="03L4GE-DE-CFD"
 1430 C\$(21)="03L4GFE-A-G"
 1440 C\$(22)="03L4FE-FDGG"
 1450 C\$(23)="03L4C<C(B-A)-GF"
 1460 C\$(24)="03L4E<C(A-FB-G"
 1470 C\$(25)="03L4C<B->C(A)->D-C"
 1480 C\$(26)="02L4B-A-B-G<C<C"
 1490 C\$(27)="02L4F<CPGFE"
 1500 C\$(28)="03L4D-E-D-C<B->D-"
 1510 C\$(29)="02L4A-R->C(B->C<C<C"
 1520 C\$(30)="02L4F<CPGA-G"
 1530 C\$(31)="03L4A-B-8>C8(B-A-GE-"
 1540 C\$(32)="02L4A-GA-FB-(B-"
 1550 C\$(33)="03L4E->E-D-C(B-A-"
 1560 C\$(34)="02L4GE-<C(A-D-B-"
 1570 C\$(35)="03L4E-D-E-CFE-"
 1580 C\$(36)="02L4D-CD-<B->E<E-"
 1590 C\$(37)="02L4A->E-A-B-A-G-"
 1600 C\$(38)="03L4FG-FE-D-F"
 1610 C\$(39)="03L4CD-E-D-E<E-"
 1620 C\$(40)="02L2A->A->C"
 1630 C\$(41)="03D1E2"
 1640 C\$(42)="03L2R2FA-"
 1650 C\$(43)="02B1>C2"
 1660 C\$(44)="04R2L2CE-"
 1670 C\$(45)="03F#1G2"
 1680 C\$(46)="03R4L4GA>C8<B-8>CC"
 1690 C\$(47)="04L8C4-B-AB->C<B4-D"
 1700 C\$(48)="04L4DC<B-G>C(A"
 1710 C\$(49)="04L4DC<B->E-D"
 1720 C\$(50)="04L4C-B->C(A)>D<D"
 1730 C\$(51)="03L4GDGAGF"
 1740 C\$(52)="03L4E-FE-DCE-"
 1750 C\$(53)="02L4B->XDCD<A"
 1760 C\$(54)="02L8BAGB>CDE-FGA-G"
 1770 C\$(55)="03L8B-AGFEDCC-B-AGF#A16G16"
 1780 C\$(56)="02L4ADA>C8<B-8>CC"
 1790 C\$(57)="03L4C2<BAGA"
 1800 C\$(58)="02L4B->XDCD<D"
 1810 C\$(59)="02L8G>DGFE-DCC<B-AB-16>C16"
 1820 C\$(60)="02L8F<CFE-DC<B-AB-AGA16B-16"
 1830 C\$(61)="02E-2R2E-2D1-"
 1840 C\$(62)="T7#V5E4Q#8#"
 1850 D\$(1)="04L2E-GA-"
 1860 D\$(2)="04B-1>C2"
 1870 D\$(3)="04A-FB-"
 1880 D\$(4)="04E-DC-"
 1890 D\$(5)="03B->DE-"
 1900 D\$(6)="04F1G2"
 1910 D\$(7)="04E-1F2"
 1920 D\$(8)="03L2B->B-A-"
 1930 D\$(9)="04L2GFE-"
 1940 D\$(10)="04L2D4E-4F<F"
 1950 D\$(11)="03L2B->DE-"
 1960 D\$(12)="04F1G2"
 1970 D\$(13)="04E-1F2"
 1980 D\$(14)="03L2B-GA-"
 1990 D\$(15)="03B-1>C2"
 2000 D\$(16)="03A-1B-2"
 2010 D\$(17)="03L2E->E-D-"
 2020 D\$(18)="04L2C<B-A-"
 2030 D\$(19)="03L2G4A-4B-B-"
 2040 D\$(20)="03L2E->E-F-"
 2050 D\$(21)="04G1A-2"
 2060 D\$(22)="04L2FDG-"
 2070 D\$(23)="04C1-"
 2080 D\$(24)="04L2C<A-B-"
 2090 D\$(25)="04C1D-2"
 2100 D\$(26)="03B-1>C2"
 2110 D\$(27)="03L2F>FE-"
 2120 D\$(28)="04L2D<C<B-"
 2130 D\$(29)="03L2A-4B-4>C<C"
 2140 D\$(30)="03F1-"
 2150 D\$(31)="03B-1>C2"
 2160 D\$(32)="03A-1B-2"
 2170 D\$(33)="03E-1-"
 2180 D\$(34)="03E-2>C2D-2"
 2190 D\$(35)="04E-1F2"
 2200 D\$(36)="04D-1E-2"
 2210 D\$(37)="03L2A->A-"
 2220 D\$(38)="04L2FE-D-"
 2230 D\$(39)="04L2C4D-4E<E-"
 2240 D\$(40)="03L2A->A->C"
 2250 D\$(41)="04D1E2"
 2260 D\$(42)="04R2F2A-2"
 2270 D\$(43)="03B1>C2"
 2280 D\$(44)="L2R2O4CE-"
 2290 D\$(45)="03F#1G2"
 2300 D\$(46)="03R4L4GAB->C2"
 2310 D\$(47)="04C4<B-8A8B-2>C2"
 2320 D\$(48)="04L2D<C<C-"
 2330 D\$(49)="04D1E-2"
 2340 D\$(50)="04C1D2"
 2350 D\$(51)="03L2GFE"
 2360 D\$(52)="04L2E-DC"
 2370 D\$(53)="03L2B-4>C4DC"
 2380 D\$(54)="03B2R2R2"
 2390 D\$(55)="03B-2R2R2"
 2400 D\$(56)="03L4ADAB->CC"
 2410 D\$(57)="04C2L4<B-AGA"
 2420 D\$(58)="03L4B->CDDC<D"
 2430 D\$(59)="03G2R2>C2"
 2440 D\$(60)="03F2R2B-2"
 2450 D\$(61)="03E-2R2E-2D1-"
 2460 OPEN "DATA.End" FOR OUTPUT AS #1
 2470 FOR I=0 TO 61 STEP 2
 2480 A\$(A)=I+1:A\$(I+1):B\$(B)=I+1:C\$(C)=I+1
 2490 WRITE #1,A\$,B\$,D\$,C\$
 2500 NEXT I:CLOSE
 C\$(I+1):D\$=D\$(I)+D\$(I+1)

リスト12

10 'Orchestral suites No.2-6 Menuet]
 20 ' Programed by D.Kobayashi in 1986
 30 'save "MUSIC7.OPE"
 40 DIM A\$(22),B\$(22),C\$(22),D\$(22)
 50 A\$(0)="T11004Q8002V9"
 60 A\$(1)="04L8B>C#DF#B4"
 70 A\$(2)="AG28G8&G4F#4E4"
 80 A\$(3)="D4C#D<B4"
 90 A\$(4)="F#EABWG"
 100 A\$(5)="F#EAD#BF#"
 110 A\$(6)="EDC#F#D<B"
 120 A\$(7)=">DC2#C"
 130 A\$(8)="04F#G#A#>C#F#4"
 140 A\$(9)="E8D2D8"
 150 A\$(10)="<AB>C#E4A"
 160 A\$(11)="GF#2#F#"
 170 A\$(12)="<BAG>C#D<B"
 180 A\$(13)="AGF#>C#D<A"
 190 A\$(14)="G4F#4E4"
 200 A\$(15)="D2-"
 210 A\$(16)=">F#ED#A4"
 220 A\$(17)="AG28G4"
 230 A\$(18)="C#DEAG4"
 240 A\$(19)="GF#2#F#"
 250 A\$(20)="<A#B>C#ED BGF#E>D<C#A#"
 260 A\$(21)="B4F#EDC#"
 270 A\$(22)="<B2-"
 280 B\$(0)="04T110Q8002.8V9"
 290 B\$(1)="04F#4B4B4"
 300 B\$(2)="B2-λL8C#D<CC#A#"
 310 B\$(3)="L4BF#B"
 320 B\$(4)="B#EG"
 330 B\$(5)="DF#"
 340 B\$(6)="<A#>C#<B"
 350 B\$(7)="A#2-"
 360 B\$(8)="L4F#>C#C#"
 370 B\$(9)="<B2-"
 380 B\$(10)="EAA"
 390 B\$(11)="A2-"
 400 B\$(12)="GDG"
 410 B\$(13)="ADA"
 420 B\$(14)="C#C#"
 430 B\$(15)="<A2-"
 440 B\$(16)=">>D8E8F#F#"
 450 B\$(17)="E2-"
 460 B\$(18)="<EA>C#"
 470 B\$(19)="D2-"
 480 B\$(20)="E<A#B#B>EG"
 490 B\$(21)="L4F#EDC#<B#A#"
 500 B\$(22)="F#2-"
 510 C\$(0)="T11004.4Q80V8"
 520 C\$(1)="04L4DF#F#"
 530 C\$(2)="E2-8E4C#G#"
 540 C\$(3)="F#4.D8E8F#B#"
 550 C\$(4)="EE"
 560 C\$(5)="F#D"
 570 C\$(6)="EF#F#"
 580 C\$(7)="F#2-"
 590 C\$(8)="C#F#F#"
 600 C\$(9)="F#2-"
 610 C\$(10)="AEE"
 620 C\$(11)="D2-"
 630 C\$(12)="DGB"
 640 C\$(13)="DF#A#"
 650 C\$(14)="DD<A"
 660 C\$(15)="F#2-"
 670 C\$(16)="BBB"
 680 C\$(17)="B2-"

```

690 C$(18)="AAA"
700 C$(19)="A2."
710 C$(20)=">GF#F#E<B>C#"
720 C$(21)="D8C#8<B>F#"
730 C$(22)="D2."
740 D$(0)="T11004Q00V6"
750 D$(1)="04L4B<B>D"
760 D$(2)="L8EF#GB>EC#<BA#4.F#4"
770 D$(3)="B4.<L8B>C#D"
780 D$(4)="L4EGC4"
790 D$(5)="DF#<B"
800 D$(6)=">C#<A#B"
810 D$(7)="F#>L8F#G#A#F#"
820 D$(8)="03L4A#F#A#"
830 D$(9)="L8B>C#DF#B<B"
840 D$(10)="L4>C#<A>C#"
850 D$(11)="L8DEF#A>D<D"
860 D$(12)="L4GBE"
870 D$(13)="F#AD"
880 D$(14)="<BGA"
890 D$(15)="L8DF#A>DF#A#"
900 D$(16)="L4<B>BD#"
910 D$(17)="L8EF#GB>E<E"
920 D$(18)="AC#<A"
930 D$(19)="L8DEF#A>DF#"
940 D$(20)="L4C#<F#BE>L8GF#EF#"
950 D$(21)="DEF#4<F#4"
960 D$(22)="B2."
970 OPEN "DATA.Ope" FOR OUTPUT AS #1
980 FOR I=0 TO 7:GOSUB *PLAY:NEXT I
990 FOR I=1 TO 22:GOSUB *PLAY:NEXT I
1000 FOR I=8 TO 22:GOSUB *PLAY:NEXT I
1010 CLOSE:END
1020 *PLAY:WRITE #1,A$(I),B$(I),D$(I),A$(I),C$(I)
1030 RETURN

```

リスト13

```

10 'Performing MUSIC1
20 'Answer "How many files (1-15) ? 6"
30 'save "list p1"
40 NEW CMD
50 OPEN "DATA A.M-1" FOR INPUT AS #1
60 OPEN "DATA B.M-1" FOR INPUT AS #2
70 OPEN "DATA C.M-1" FOR INPUT AS #3
80 OPEN "DATA D.M-1" FOR INPUT AS #4
90 OPEN "DATA E.M-1" FOR INPUT AS #5
100 OPEN "DATA F.M-1" FOR INPUT AS #6
110 IF EOF(1) THEN CLOSE:END
120 INPUT #1,A$:INPUT #2,B$:INPUT #3,C$
130 INPUT #4,D$:INPUT #5,E$:INPUT #6,F$
140 CMD PLAY #2,A$,B$,D$,C$,E$,F$
150 GOTO 110

```

リスト14

```

10 'Performing MUSIC2~7
20 '
30 'save "list p2"
40 NEW CMD
50 DIM FILE$(6)
60 CLS:PRINT "Select Music Number (1-6)"
70 FOR I=1 TO 6:READ FILE$:PRINT I:--- " :FILE$(I):NEXT I
80 I$=INKEY$:IF I$="" GOTO 80
90 I=VAL(I$)
100 IF I>6 OR I<1 GOTO 80
110 OPEN FILE$(I) FOR INPUT AS #1
120 ON I GOTO *P1,*P2,*P2,*P2,*P1,*P3
130 *P1
140 IF EOF(1) THEN CLOSE:END
150 INPUT #1,A$,B$,D$,C$
160 CMD PLAY #2,A$,B$,D$,C$
170 GOTO 140
180 *P2
190 INPUT #1,A$,B$,C$,D$,E$,F$
200 CMD PLAY #2,A$,B$,D$,C$,E$,F$:CMD PLAY #2,"r1"
210 IF EOF(1) THEN CLOSE:END
220 INPUT #1,A$,B$,C$,D$,E$,F$
230 CMD PLAY #2,A$,B$,D$,C$,E$,F$
240 GOTO 210
250 *P3
260 IF EOF(1) THEN CLOSE:END
270 INPUT #1,A$,B$,C$,D$,E$
280 CMD PLAY #2,A$,B$,C$,D$,E$
290 GOTO 260
300 DATA DATA.M-2,DATA.M-3,DATA.M-4,DATA.M-5,DATA.End,DATA.Ope

```


斜めスクロールスーパグラフィックスシューティングゲーム

AZON-VA

前号で88VA用グラフィックツール「お絵描きくん」を作った筆者がそのノウハウを生かして、ゲームセンター顔負けの本格的シューティングゲームをすべてBASICで作ってしまいました。88VAならではのグラフィックスは圧巻です(「88WINDOW」参照)。また、次号より筆者のゲーム作成講座もスタートします。みなさんの質問を受けつけますので、どしどし送ってください。

大島 篤

オールBASICのシューティングゲームだぞ!

前号で掲載した88VA専用グラフィックツール「お絵描きくん」を知っていますか? (知らない人は「Oh!88」No.158ページ参照) あれば6つのサイズのペンと256色の色を使うだけのごく簡単なものでしたが、そのV3-BASICの威力でメニュー表示のウィンドー機能やグラフィック表示がなんとすべてBASICレベルでできてしまうのに驚かされたものです。

この88VAの能力をもってすれば、ゲームの「ゼビウス」くらいはすべてBASICで作れるのではないかと、とずっと考えていました。今回はさらにそれを確かめるため、敵機がパンパン出てきて、弾をビシバシ撃ってきて、地形が斜めにスクロールして、地上の基地からも弾がドンドコ発射されるというゲームを作ってみることにしました。つまり「ゼビウス」の斜めスクロール版のようなものです。

88VAのV3-BASICのすばらしさをよく知らない人が聞いたら、

「そんなもん BASIC で作れるわけないじゃん!」

と、一笑するかもしれません。ところがどっこい! 作ってみたら、できてしまいました!

V3-BASICの底力

リアルタイムシューティングゲームを作る場合、キャラクタの移動ルーチンと、地形のスクロールルーチンを、いかに速く作るのが重要となります。この2つの仕事は、従来のパソコンにとっては非常に重荷で、ゲームプログラムの全体の速度を100とす

れば、その90以上をこの作業のために費やしてしまっていると思われまふ。

ご存じとは思いますが、88VAはスプライト機能を持っています。これでキャラクタを高速に動かすことができます。しかも、スプライトパターンのサイズが可変です。

また、広大な画面領域を設定しておき、その一部を表示するフレームバッファ機能があるため、もともと地形を描いておき、非常に高速にスムーズスクロールさせることもできます。なんと毎秒数百ドットのスムーズスクロールが可能なのです。

というわけで、ゲームプログラム中の90の要素は、何もしなくてできてしまうのです。プログラムは、残りの10の要素に力を注げばいいのです。

ところが、いくらスプライトの表示が高速にできても、スプライトの表示座標をBASICで右に10ドット進ませるんだ、下に20ドットだ、といちいち制御していたのでは、やはり遅くてゲームになりません。

88VAの底力はここからです。スプライトマクロ機能というのがあるので、これは、

「5番のスプライトは始めは10、10の座標に出現させて、300、200まで移動して、そこから1番のスプライトの位置に進む」などといった動きを前もって決めておき、命令を一発使うだけであとはBASICプログラムと並列処理で勝手にスプライトが動いてくれるのです。

これで、100のうち95はできてしまったことになります。これなら残りの5の仕事くらいは、BASICでやっても大丈夫です。

本当に、88VAはすごいです。X68000のほうがすごいとかいう人もいますが、X68000はスプライトマクロ機能がないので

BASICだけで高度なゲームを作るのは困難かと思ひます。

なお、今回のプログラムには、スプライトや地形データをデザインするためのツールの作成およびデザイン作業まで含めて、1週間程度しかかかっていません。それも、けっこう遊びながら睡眠も十分取ってます。

こんなに簡単に作ってしまうのは、88VAの能力のたまものです。

ゲームの概要

このゲームはマウスまたはジョイスティックで操作します。

プレイヤー機を操って、飛来してくる敵をミサイル(左ボタン)で打ち落とし、地上物をプラスター(右ボタン)で破壊してください。

このゲームは、ステージ1しかデータがありません。

リスト3やリスト4を見てもらえば分かります。データ量が非常に多いので、これ以上ステージデータを掲載することが難しい、入力の手間も大変だからです。

というわけで、ステージ1をクリアすると、再びステージ1の先頭から始まります。

地上の攻撃可能な建造物や兵器を、すべて破壊することがゲームの最終目的です。

友人のデザイナー風谷氏による画面の美しさは、ゲームセンターのマシンに勝るとも劣りません。マップの基本レイアウトはS氏の設計です。BGMがないのは、私の力不足です。御容赦。

プログラムの入力方法

まず、最初にリスト1を入力して、RUNしてください。これは、キャラクタのスプライトパターンをディスクに登録するプロ

グラムです。

次に、リスト2を入力します。

MON ☺

でマシン語モニタに入って、

K7000:0 ☺

とすれば、スクリーンエディット形式のマシン語入力モードとなります。といっても、リスト2はマシン語ではなく、地上図形のマップデータです。

入力できたら、

Q ☺

でBASICにもどることができます。BASICにもどったら、

DEF SEG=&H7000 ☺

BSAVE "STAGEL.MAP", 0,

2016 ☺

として、入力したデータをディスクにセーブしてください。

次に、リスト3を入力します。マシン語モニタで、

K7000:1000 ☺

として、入力し始めてください。入力できたら、BASICにもどって、

DEF SEG=&H7000 ☺

BSAVE "GROUND.PAT", &H1000,

&H1300 ☺

として、ディスクにセーブします。これは地上のマップデータです。なお、マシン語を入力する際はVA用チェックサムプログラム(リストA)を入力してください。リスト4が、ゲームプログラムです。ジョイスティックを使うときは、プログラム1090行を次のようにしてください。

1090 DEVICE=1

プログラムを入力して、ディスクに

SAVE "AZON" ☺

としてセーブしてください。

RUN "AZON" ☺

とすれば、ゲームが始まります。

次号予告です

実は、モビルアタッカーという浮上走行メカをメインキャラクターとしたゲームを作る予定で、実際に制作開始したのですが、それは時間的に間に合いそうもないので、今回のようなゲームに変えたのです。モビルアタッカーは今度作るつもりですから、お楽しみに。

今回のゲーム作成で得たノウハウ、ゲーム作成のコツを、次号で「ゲーム作成講座」を開き、誌上公開したいと思います。

そこで、みなさんからの質問、マップの作り方、ストーリー作成のコツ、何でも結構ですのでどんな「Oh!88 ゲーム作成講座係」までお寄せください。

X68000を持っている友達にこのゲームを見せて、BASIC オンラインであることを告げて驚かせよう!

というわけで、88VAは無敵だった!

「VAクラブ」のお知らせ

「Oh!88」では、筆者の大島篤さんを中心に「VAクラブ」を結成したいと思います。「マニュアルには書いていないけど、こんな命令を発見した」「この機能はどう使えばいいの?」など本誌読上でVAユーザーの情報交換の場を作りませんか?

プログラム作成上の悩み、VAに対する感想などなんでも結構です。アンケート葉書きとは別に、官製葉書きに書いて「Oh!88 VAクラブ係」までお送りください。質問には本誌にて大島篤さんが回答します。なお、次号発売は4月28日の予定です。どしどしお葉書きお寄せください。VAユーザー集まれ!

リストA

```
100 ' PC-88VA用チェックサムプログラム by 大島篤
110 '
120 INPUT "セグメントアドレス";SEG$
130 INPUT "開始オフセットアドレス";SOFF$
140 INPUT "終了オフセットアドレス";EOFF$
150 SEG=VAL("&"&SEG$): DEF SEG=SEG
160 SOFF=VAL("&"&SOFF$): SOFF=INT(SOFF/16)*16: IF SOFF<0 THEN SOFF=SOFF+65536:
170 EOFF=VAL("&"&EOFF$): IF EOFF<0 THEN EOFF=EOFF+65536:
180 INPUT "プリンタに出力しますか(Y/N)";PS$
190 IF PS$="Y" OR PS$="y" THEN OPEN "LPT:" FOR OUTPUT AS #1 ELSE OPEN "SCRN:" FO
R OUTPUT AS #1
200 '
210 DEF FNH$(N)=RIGHT$("0"+HEX$(N),2)
220 ADDR=SOFF
230 DIM VSUM(15)
240 WHILE NOT (ADDR>EOFF)
250 PRINT #1,"SEG:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM"
260 FOR T=0 TO 15: VSUM(T)=0: NEXT
270 FOR TX=0 TO 15
280 PRINT #1,SEG$,";";
290 PRINT #1,RIGHT$("0000"+HEX$(ADDR),4);" ";
300 HSUM=0
310 FOR TX=0 TO 15
320 DAT=PEEK(ADDR): ADDR=ADDR+1
330 HSUM=HSUM+DAT: VSUM(TX)=VSUM(TX)+DAT
340 PRINT #1,FNH$(DAT);
350 IF TX=7 THEN PRINT #1,"-"; ELSE PRINT #1," ";
360 NEXT TX
370 PRINT #1,";";FNH$(HSUM)
380 NEXT
390 PRINT #1,STRING$(63,"-")
400 TSUM=0
410 PRINT #1," VSUM ";
420 FOR T=0 TO 15
430 PRINT #1,FNH$(VSUM(T));" ";
440 TSUM=TSUM+VSUM(T)
450 NEXT
460 PRINT #1,";";FNH$(TSUM)
470 '
480 IF PS$="N" OR PS$="n" THEN PRINT "何かキーを押すと次を表示します": AS=INPUT$
(1)
490 PRINT #1,: PRINT #1,: PRINT #1,
500 WEND
510 PRINT "終わります"
```

```
0000 ' スプライトパターン定義プログラム&データ
1010
1020 DEFINT A-Z
1030 SCREEN WIDTH 5,640,4: SCREEN ,0
1040 DIM GX(10)
1050 QX=0
1060 *LOOP
1070 READ P: IF P<0 THEN *SVPAT
1080 READ X,Y
1090 NEED.BYTE=(X\2)\Y+4: MAX.A=NEED.BYTE*2+1
1100 ERASE GX: DIM GX(MAX.A)
1110 GX(0)=X: GX(1)=Y
1120 FOR T=2 TO MAX.A: READ DS: GX(T)=VAL("&H"&DS): NEXT
1130 PUT (GX(0),GX,PSET: GX=GX+*8
1140 PUT PATTERN F,GX,4
1150 GOTO *LOOP
1160 *SVPAT
1170 SAVE PATTERN "GAME.SPR",ALL
1180 PRINT "スプライトパターンデータ作成完了"
1190 END
1200
1210 ' プレイヤー機
1220 DATA 1,64,32
1230 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,F7F7
1240 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,FFFF,F7FF
1250 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,00FF,0000,FFFF,7777,B0FB
1260 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,1111,1111,7777,7777,FF77,00B0
1270 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0F00,11F1,1111,1111,9919,BF97,7777,B0FF,000A
1280 DATA 0000,0000,0000,0000,0000,0800,7FFF,8FFF,7787,9F7F,1191,9999,9999,FB87,7777,00FB,0000
1290 DATA 0000,0000,0000,0000,0000,7708,7777,7777,8FFF,9999,9999,B7BB,7777,7777,7777,7777,00B0,0000
1300 DATA 0000,0000,0000,0800,7777,7777,777E,FF77,8888,99FB,B9B9,77B7,B888,B778,0000,0000,0000
1310 DATA 0000,0000,0000,7708,7777,F877,F877,88FF,8888,B8BF,B7BB,8877,B8BB,0BBB,0070,0000
1320 DATA 0000,7707,8877,7777,7777,7778,B777,8888,8888,7788,7777,8888,88BB,0000,0007,0000
1330 DATA 7700,7777,7788,7777,7777,7787,FF77,9BBB,8888,8888,8888,B8BB,0000,0000,700B,0000
1340 DATA 7777,8877,7777,7777,8877,7777,887F,8888,8888,B7BB,B8BB,B7BB,0000,0000,700B,0000
1350 DATA 0700,7778,8878,7777,7778,E777,8888,88BB,8888,77B7,B7B7,B0B7,0000,0000,0000,0000
1360 DATA 0000,BBFF,8828,7788,7787,887F,8888,88BB,87BB,7777,B777,B777,70B0,0000,0000,0000,0000
1370 DATA 0000,FFFF,2A2B,8828,7777,88FB,8888,88BB,7787,7777,7777,87B0,0000,0000,0000,0000,0000
1380 DATA 0200,FFFF,2AFF,87AB,F877,8888,8888,B7BB,7777,7777,7777,8877,0070,0000,0000,0000,0000
1390 DATA 0200,FFFF,FFFF,F7BB,887F,8888,8888,7777,7777,7777,7777,7777,0077,0000,0000,0000,0000
1400 DATA AA02,FFFF,FFFF,FFFF,8888,88BB,B7BB,7777,7777,7777,7777,7777,887F,K0BF,0000,0000,0000
1410 DATA ABAB,FFFF,FFFF,FFBB,8888,88BB,B7BB,7777,7777,7777,7777,7777,8888,8888,0000,0000,0000
1420 DATA BZAB,FFFF,FFFF,B7FB,B8BB,88BB,B7BB,7777,7777,7777,887F,8888,88BB,0000,0000,0000,0000
1430 DATA B2BA,AABB,FA7B,77B7,77B7,88BB,88BB,AA7B,7777,7777,887F,8888,8888,0000,0000,0000,0000
1440 DATA 0200,AAAF,77AB,77B7,77B7,88BB,88BB,AAAA,7777,7777,887F,8888,8888,0000,0000,0000,0000
1450 DATA 0800,ABA2,BBBB,8877,88BB,AAA2,ABAB,7788,7777,8888,8888,B0BB,0000,0000,0000,0000,0000
1460 DATA 0000,B00B,BB0B,88BB,22BB,ABAA,BZAB,87B8,887F,8888,8888,0000,0000,0000,0000,0000,0000
1470 DATA 0000,0000,88BB,88BB,AA22,ABAA,BABA,22B2,7FAA,8888,8888,B077,0000,0000,0000,0000,0000
1480 DATA BB8B,B8BB,88BB,88AA,BZAB,22B2,BSAA,8888,7788,0077,0000,0000,0000,0000,0000,0000,0000
1490 DATA 0000,0B0B,BBBB,ABBB,ABBA,22B2,AAAA,8888,8888,7777,0000,0000,0000,0000,0000,0000,0000
1500 DATA 0000,0B0B,BOB0,0A00,A2BB,AA22,BAAA,BBBB,7788,7777,7777,0000,0000,0000,0000,0000,0000
1510 DATA 0000,0B0B,00B0,8800,A2BB,ABAA,BABA,BBBB,07BB,7777,7777,0000,0000,0000,0000,0000,0000
1520 DATA 0000,0000,0000,0008,AAAA,AAAB,BBBA,0000,0000,7777,7777,0000,0000,0000,0000,0000,0000
1530 DATA 0000,0000,0000,0000,8800,AAAB,00B0,0000,0000,7707,0077,0000,0000,0000,0000,0000,0000
1540 DATA 0000,0000,0000,0000,0008,ABAA,0000,0000,0000,7700,0000,0000,0000,0000,0000,0000,0000
1550 DATA 0000,0000
1560 ' プラステア組拳
1570 DATA 10,16,8
1580 DATA 0077,0000,0000,7700
1590 DATA 7700,0000,0000,0077
1600 DATA 0000,7777,7777,0000
1610 DATA 0000,2207,7022,0000
1620 DATA 0000,2207,7022,0000
1630 DATA 0000,7777,7777,0000
1640 DATA 7700,0000,0000,0077
1650 DATA 0077,0000,0000,7700
1660 DATA 0000,0000
1670 ' プラステア固定
1680 DATA 11,16,8
1690 DATA 0000,2200,00B0,2B00
1700 DATA 0000,2222,00B0,002B
1710 DATA 2200,2222,00B0,0000
1720 DATA 2222,2222,BBBB,BB77
1730 DATA BBBB,BBBB,2222,2222
1740 DATA 0000,0B00,2222,BB22
1750 DATA 0000,0B2B,2222,00BB
1760 DATA 2B00,00B0,BB22,0000
1770 DATA 0000,0000
1780 ' 対空サイル
1790 DATA 12,16,8
1800 DATA 0000,0000,0000,2202
1810 DATA 0000,0000,62B8,2022
1820 DATA 0000,60B6,776D,0022
1830 DATA 06F0,60F0,0077,0000
1840 DATA F6F0,77F0,F600,0060
1850 DATA F000,007F,6600,0000
1860 DATA 7700,0F06,00F0,0000
1870 DATA 0077,0F60,F000,0000
1880 DATA 0000,0000
1890 ' プラステア
1900 DATA 13,16,8
1910 DATA 0000,0000,0000,0000
1920 DATA 7000,0000,0000,0000
1930 DATA B0AD,7070,0600,2066
1940 DATA 200B,0702,7607,2266
1950 DATA BA20,AAZA,7707,2277
1960 DATA 00DA,ZAZA,ATAA,2077
1970 DATA OA07,AAK7,0000,0000
```



```

1980 DATA 0000,0000,0000,0000
1990 DATA 0000,0000
2000 ' 爆機
2010 DATA 20,32,20
2020 DATA 0077,0000,0070,7700,0770,0060,0000,0000
2030 DATA 0607,7060,7070,7000,0600,7007,0000,0700
2040 DATA 0000,6060,0000,0060,0600,0600,7700,0000
2050 DATA 0000,6666,0760,6060,0000,6706,0000,0000
2060 DATA 7070,6606,0760,60B2,800B,6B66,00B0,0060
2070 DATA 0000,6007,0B66,660B,66BB,0B06,6600,0700
2080 DATA 0000,8B00,6B06,6BBB,00BB,8066,0006,0000
2090 DATA 0007,0000,6600,00BB,86B0,8666,0060,0000
2100 DATA 0000,0200,00B0,86BB,6677,6666,800B,0000
2110 DATA 7600,0666,0766,777B,7B76,62B6,07BB,6066
2120 DATA 6666,8BB6,6066,7007,7777,6626,7022,0770
2130 DATA 7000,0200,807B,0000,72B7,80BB,0000,0007
2140 DATA 0000,0000,66BB,06B0,6BBB,6006,00B0,0007
2150 DATA 0000,0070,66B0,86BB,87BB,66BB,0066,0000
2160 DATA 0700,0707,6802,86BB,8B06,06BB,6660,0000
2170 DATA 0000,7700,6B26,00BB,6260,8B0B,0600,0006
2180 DATA 0000,0000,0B62,0700,6007,8022,7000,6000
2190 DATA 0000,0000,0070,0760,6000,0000,0000,0607
2200 DATA 0000,0000,0070,0070,7600,0700,0000,0000
2210 DATA 7700,0000,0600,7000,7007,0070,0000,0000
2220 DATA 0000,0000
2230 ' 回転UFO 1
2240 DATA 30,32,16
2250 DATA 0000,0000,FF00,FFFF,FFFF,0088,0000,0000
2260 DATA 0000,0700,7F77,FFFF,88FF,8888,00B0,0000
2270 DATA 0000,7707,FFFF,FFFF,8888,8888,8088,0000
2280 DATA 0700,7F77,FFF7,88FF,8888,8888,8888,00B0
2290 DATA 7700,FF77,FF7F,8FBF,8888,8888,88BB,00BB
2300 DATA 7707,7F7F,FFFF,888F,8888,AA88,8888,80BB
2310 DATA 7777,FFFF,88FF,8FBF,8A88,AAAA,88BB,88BB
2320 DATA 7777,FFF7,88FF,8888,AA8A,AAAA,88BB,88BB
2330 DATA 7F77,FFFF,8FBF,8AF8,AAAA,AAAA,88BB,88BB
2340 DATA 7F77,FF7F,8FBF,8888,AA8A,AAAA,88BB,88BB
2350 DATA 7F07,88FF,8FBF,8888,8A88,AAAA,88BB,80BB
2360 DATA 7700,FFFF,88FF,8888,8888,AA8A,88BB,00BB
2370 DATA 0700,8F7F,8FBF,888F,8888,8A88,88BB,00B0
2380 DATA 0000,8FB8,8FBF,88BB,88BB,88BB,88BB,0000
2390 DATA 0000,8B00,8888,8888,88BB,88BB,00B0,0000
2400 DATA 0000,0000,8B00,88BB,88BB,00BB,0000,0000
2410 DATA 0000,0000
2420 ' 回転UFO 2
2430 DATA 31,32,16
2440 DATA 0000,0000,FF00,FFFF,FFFF,0088,0000,0000
2450 DATA 0000,0700,7F77,FFFF,88FF,8888,00B0,0000
2460 DATA 0000,7707,FFFF,FFFF,8888,8888,8088,0000
2470 DATA 0700,7F77,FFF7,88FF,8888,8888,8888,00B0
2480 DATA 7700,FF77,FF7F,8FBF,8888,8888,88BB,00BB
2490 DATA 7707,7F7F,2222,88FF,8888,8888,8888,80BB
2500 DATA 7777,FFFF,2222,2222,8888,8888,8888,88BB
2510 DATA 7777,FFF7,2222,2222,8828,8888,88BB,88BB
2520 DATA 7F77,FFFF,2222,2822,8888,8888,88BB,88BB
2530 DATA 7F77,FF7F,2222,8828,8888,8888,88BB,88BB
2540 DATA 7F07,88FF,2822,8888,8888,88BB,88BB,80BB
2550 DATA 7700,FFFF,8F22,8888,8888,88BB,88BB,00BB
2560 DATA 0700,8F7F,8828,8888,88BB,88BB,88BB,00B0
2570 DATA 0000,8FB8,8F2F,88BB,88BB,88BB,88BB,0000
2580 DATA 0000,8B00,8888,88BB,88BB,88BB,00B0,0000
2590 DATA 0000,0000,8B00,88BB,88BB,00BB,0000,0000
2600 DATA 0000,0000
2610 ' 回転UFO 3
2620 DATA 32,32,16
2630 DATA 0000,0000,FF00,FFFF,FFFF,0088,0000,0000
2640 DATA 0000,0700,7F77,FFFF,88FF,8888,00B0,0000
2650 DATA 0000,7207,FF2F,FFFF,8888,8888,8088,0000
2660 DATA 0700,2272,FF2F,88FF,8888,AA88,88AA,00B0
2670 DATA 7200,2222,FF22,88FF,8888,8888,AAAA,00BB
2680 DATA 2207,2F22,FFFF,88FF,8888,8888,88AA,80AB
2690 DATA 2272,FFFF,FFFF,88FB,8888,8888,88BB,88AA
2700 DATA 2772,FFF7,FFFF,88FB,8888,8888,88BB,88BB
2710 DATA 7F22,FFFF,FFFF,8888,8888,8888,88BB,88BB
2720 DATA 7F27,FF77,88FF,8888,8888,8888,88BB,88BB
2730 DATA 7F07,88FF,88FF,8888,8888,88BB,88BB,80BB
2740 DATA 7700,FFFF,88FF,8888,8888,88BB,88BB,00BB
2750 DATA 7700,8F7F,88FB,8888,88BB,88BB,88BB,00B0
2760 DATA 0007,8FB8,88FF,8888,88BB,88BB,88BB,0000
2770 DATA 0000,8B00,8888,88BB,88BB,88BB,00B0,0000
2780 DATA 0000,0000,8B00,88BB,88BB,00BB,0000,0000
2790 DATA 0000,0000
2800 ' UFO 4 2
2810 DATA 40,32,20
2820 DATA 0000,0000,0800,8888,8888,00B0,0000,0000
2830 DATA 0000,0800,8888,88BB,88BB,8888,00B0,0000
2840 DATA 0000,8F08,88BB,88BB,88BB,88BB,8088,0000
2850 DATA 0000,7888,88BB,22B2,2B22,88BB,8888,0000
2860 DATA 0800,888F,88BB,22BB,2B22,88BB,8888,00B0
2870 DATA 8800,88FF,8B2B,2B2B,8B2B,8B2B,88FF,00BB
2880 DATA 8F08,88FF,8B22,88BB,88BB,22BB,88FF,80BB
2890 DATA 8F08,88FF,2B22,88BB,88BB,22B2,88FF,80BB
2900 DATA 888F,8888,2B22,88BB,88BB,28B2,8878,88BB
2910 DATA 8828,88BB,88BB,88BB,88BB,78BB,88BB,88BB
2920 DATA 888F,8888,8888,2288,8F22,8888,88BB,22BB
2930 DATA 8B28,88BB,88BB,8888,8888,8888,88BB,88BB
2940 DATA 88BB,88BB,88BB,88BB,88BB,88BB,88BB,22BB
2950 DATA 8B0B,88BB,88BB,88BB,88BB,88BB,88BB,88BB

```

```

2960 DATA BB0B,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BOBB
2970 DATA BB0D,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BOBB
2980 DATA BB0D,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BBBB,BOBB
2990 DATA 0000,BBBB,BB2B,BBBB,BBBB,BBBB,BB2B,BBBB,0000
3000 DATA 0000,BB0D,BB2B,BBBB,BBBB,BBBB,BB2B,00BB,0000
3010 DATA 0000,0000,BB0D,BBBB,BBBB,BBBB,00BB,0000,0000
3020 DATA 0000,0000
3030 ' U F C その3
3040 DATA 50,32,20
3050 DATA FFOF,FFFF,FFFF,FFFF,FFFF,FFFF,00B0,0000,0000
3060 DATA FF7F,FFFF,FFFF,FFFF,FBFF,00B0,0000,0000,0000
3070 DATA FB7F,BBBB,BBBB,BBBB,BBFB,6000,0000,0000,0000
3080 DATA FB7F,11B1,1111,FF1F,B0BB,00FF,0000,0000,0000
3090 DATA FB7F,11B1,B711,FFFF,FFFF,F9FF,0000,0000,0000
3100 DATA FB7F,11B1,B71F,F7FF,1F11,F9FF,00B0,0000,0000
3110 DATA FB7F,1FB1,B7FF,71F1,B111,F9BF,60B0,0000,0000
3120 DATA FB7F,FBFB,B7FB,11F1,BB11,F9BB,00BF,7B00
3130 DATA FF7F,BBFB,B7BB,11F1,1B1B,F9BB,00BB,BB7F
3140 DATA FF7F,BBFB,B700,11FF,BB11,F9BB,7FBF,BBFF
3150 DATA FB7F,00BB,B70F,FFFF,BBBB,F9BF,FBFB,FBF7
3160 DATA BB7B,0F00,B7F0,FFFF,FFFF,FFFF,FBBF,FBF7
3170 DATA 007B,F006,7900,9999,9999,9999,9999,B1BB,FBF7
3180 DATA 0000,0000,BB0D,BBBB,BBBB,BBBB,BBBB,11B1,FBF7
3190 DATA 0000,0000,0B0D,BBFB,7BBB,B1FB,1111,FBF7
3200 DATA 0000,0000,0F00,00F0,FB7F,11B1,1111,FBF7
3210 DATA 0000,0000,F006,7F00,B1FB,1111,1111,FBF7
3220 DATA 0000,0000,0000,FF7F,FFFF,FFFF,FFFF,FBFF
3230 DATA 0000,0000,7F00,FFFF,FFFF,FFFF,FFFF,FBFF
3240 DATA 0000,0000,BB0B,BBBB,BBBB,BBBB,BBBB,BBBB
3250 DATA 0000,0000
3260 ' 縦線機右バンク
3270 DATA 60,40,24
3280 DATA 0000,0000,0000,0000,0000,0000,4400,2202,2222,4422,4C44
3290 DATA 0000,0000,0000,0000,0000,0000,4404,4444,4444,4444,6C44
3300 DATA 0000,0000,0000,0000,0000,0000,DD00,4444,4474,4444,4444,CC6C
3310 DATA 0000,0000,0000,0000,0000,0000,D4DD,7744,4777,4474,4644,CC6C
3320 DATA 0000,0000,0000,0000,0000,0000,DDDD,4400,4447,7747,4444,CC46,CC6C
3330 DATA 0000,0000,6D00,0000,0000,4404,4444,4444,6644,CCCC,C0CC
3340 DATA 0044,0000,0000,0000,DDDD,4444,4444,4444,CC6C,CCCC,CACC
3350 DATA 0000,0000,0000,0000,045D,BB4B,BBBB,BBBB,CCCC,AAFC
3360 DATA 0000,0000,0000,4D00,BBBB,757B,BC5B,CFCC,CCCC,AAFC
3370 DATA 0000,0000,0000,BB04,77B7,5577,BCDB,CFCC,CCCC,AAFC
3380 DATA 0000,0000,0000,B74B,7577,D155,BC1B,CFCC,CCCC,AAFC
3390 DATA 0000,0000,0000,77BB,5575,11D1,BC9B,CFCC,CCCC,AAFC
3400 DATA 0000,0000,4B00,75B7,D155,9911,BC9B,FFCC,CFCC,AAFC
3410 DATA 0000,0000,BB04,5577,115D,9999,BC9B,FFCC,CCCC,AAFC
3420 DATA 0000,0000,B74B,5D75,9911,9999,BC9B,FFCF,CFCF,AAFC
3430 DATA 0000,0000,0000,77BB,115D,9919,9999,BC9B,FFCF,CFCF,AAFC
3440 DATA 0000,4B00,5DB7,1911,9999,9999,BCBB,CCCC,CCCC,C0CC
3450 DATA 0000,BB04,117D,9919,9999,BB9B,BCBB,CCCC,CCCC,C0CC
3460 DATA 0000,B74B,1911,9999,9999,BBBB,CCBC,CCCC,CCCC,C0DD
3470 DATA 0400,71BB,9999,9999,BBBB,CCBB,CCCC,0DCC,0D00,00D0
3480 DATA 4B00,19B9,9999,BBBB,CCBC,CCCC,0C0D,0D00,0000,0000
3490 DATA B104,BB19,BBBB,CCCC,0C0C,0000,0000,006D,00DD,0000
3500 DATA BB4C,CBBB,CCCC,0C0C,0000,0000,0000,0000,0000,0000
3510 DATA CC6C,CCCC,0000,0000,0000,0000,0000,6D00,0000,0000
3520 DATA 0000,0000
3530 ' 縦線機左バンク
3540 DATA 61,40,24
3550 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,C006,0000
3560 DATA 0000,0000,0000,0000,0000,0000,0000,0600,CCCC,0000
3570 DATA 0000,0000,0000,0000,0000,0000,0000,0000,C008,CBCC,0000
3580 DATA 0000,0000,0000,0000,0000,0000,0600,CCCC,B2CC,0020
3590 DATA 0000,0000,0000,0000,0000,0000,0000,6C00,CCCC,C2CC,0020
3600 DATA 0000,0000,0000,0000,0000,0000,0000,CF6C,CCCC,CBCC,0022
3610 DATA 0C00,0000,0000,0000,0000,0000,6C00,CF6C,CCCC,CBCC,0022
3620 DATA 0000,0000,0000,0000,0000,0000,CC0D,CCCC,CF6C,CF6C,20B2
3630 DATA 0000,0000,0000,0000,0000,0000,BC55,CCCC,CF6C,FCFF,22C2
3640 DATA 0000,0000,0000,0000,0500,BC99,CFCC,CF6C,FCFF,2C0B
3650 DATA 0000,0000,0000,0000,0905,BB99,CFCC,CF6C,CCCC,CCCC
3660 DATA 0000,0000,0000,0000,995D,BB9B,CFCC,FFFF,CF6C,CCCC,CCCC
3670 DATA 0000,0000,0000,5D00,9999,BB9B,CF6C,CF6C,CCCC,BB9B
3680 DATA 0000,0000,0000,9905,9999,BBBB,CCBC,CCCC,CB9C,DBCD
3690 DATA 0000,0000,0000,9959,9999,BBBB,CCBB,BCCC,DB9C,B0DB
3700 DATA 0000,0000,5D00,9999,BB99,BBBB,CCBC,CB9C,DB9C,B0DB
3710 DATA 0000,0000,9905,9999,BBBB,CCBC,BC9C,BB9B,B0DB,0000
3720 DATA 0B00,0000,99D9,BB9B,CCBB,CB9C,BB9C,BB9B,00B0,0000
3730 DATA 0000,5B00,9B99,CCBB,BCCC,BB9C,BBBB,B0DB,0000,0000
3740 DATA 0000,9905,BB9B,CCBC,CB9C,BBBB,BB9B,B0DB,00B0,0000
3750 DATA 0500,9BD9,CCBC,BC9C,BBBB,00BB,B06B,0000,0000,0000
3760 DATA 5900,CC9B,CB9C,BBBB,B0BB,0000,0000,0000,0000,0000
3770 DATA BC0C,BCCC,BBBB,BBBB,0000,0000,0000,0000,0000,0000
3780 DATA CB6C,BB9C,B0BB,0000,0000,0000,0000,0000,0000,0000
3790 DATA 0000,0000
3800 ' 爆撃機
3810 DATA 70,40,32
3820 DATA 0000,0000,0000,0000,0000,2202,2222,2222,2222,7722,0000
3830 DATA 0000,0000,0000,0000,0200,22F2,F2F2,2222,2222,BB77,0000
3840 DATA 0000,0000,0000,2F02,F2F2,22F2,8888,778B,BBBB,0010
3850 DATA 0000,0000,0200,F2F2,F2F2,28F2,BBBB,BBBB,11BB,0711
3860 DATA 0000,0000,2202,2F1F,F2F2,8B22,BBBB,BBBB,1111,7A17
3870 DATA 0000,0200,1121,2FF2,F2F2,BB28,11B7,1111,1711,AA7A
3880 DATA 0000,2102,1211,2222,2222,BBBB,7111,1111,7A17,AAAA
3890 DATA 0200,22F2,2222,2222,2822,B1BB,1771,1111,AA1F,AAAA
3900 DATA 2202,2022,2202,2222,8B22,11BB,1117,1117,AA1F,AAAA
3910 DATA 0000,00AA,2222,2222,2222,17B1,7111,1117,AA1F,AAAA
3920 DATA AA00,0000,FFFF,FFFF,2222,1121,1771,7111,AA1F,AAAA
3930 DATA 0006,AA00,BBBB,BBBB,22B2,1122,1111,1111,AA1F,AAAA

```

```

3940 DATA 0000,00AA,22B2,2A22,22BB,FF2A,FFFF,FFFF,AAFF,AAAA
3950 DATA 0600,0200,2222,BABA,2ARB,AAAA,AAAA,AAAA,AAAA,AAAA
3960 DATA 0000,2500,AA51,BABA,AABB,AAAA,AAAA,AAAA,AAAA,AAAA
3970 DATA 0000,7502,AA11,BABA,AABB,AAAA,AAAA,FFAA,AFAA,AAFA
3980 DATA 0000,1127,AA99,BABA,AABB,AAAA,AAAA,AFAA,FFFA,AAAA
3990 DATA 0200,1171,AA99,BABA,AABB,AAAA,AAAA,AAAA,FAFF,AAAA
4000 DATA 2700,9911,AA99,BABA,AABB,AAAA,AAAA,AAAA,FAFF,AAAA
4010 DATA 7902,9919,AA9A,BABA,AABB,BAAA,BABA,AAAA,AAAF,AAFA
4020 DATA AAF A,AAAA,AAAA,BABA,AABB,BAAA,BABA,AFAA,AAFA,AAFF
4030 DATA 99B9,9999,9999,9999,AA9B,BAAA,BABA,AAAA,AAAA,AAAA
4040 DATA BB0B,BBBB,BBBB,AABB,BAAA,BABA,AAAA,AAAA,AAAA
4050 DATA 0000,0000,0000,0000,9909,9999,9999,9999,9999,9999
4060 DATA 0000,0000,0000,0000,BB0B,BBBB,BBBB,BBBB,BBBB
4070 DATA 0000,0000,0000,0000,0000,000A,2B22,1111,2212,BB22
4080 DATA 0000,0000,0000,0000,0000,AA00,0000,2B22,1211,2222,0BBB
4090 DATA 0000,0000,0000,0000,000A,AA00,2B22,2212,BB22,0000
4100 DATA 0000,0000,0000,0000,0000,0000,00AA,2B22,0000,0000
4110 DATA 0000,0000,0000,0000,0000,0000,0000,2222,BB22,0000
4120 DATA 0000,0000,0000,0000,0000,0000,0000,2222,0BBB,0000,0000
4130 DATA 0000,0000,0000,0000,0000,0000,0000,BB22,0000,0000,0000
4140 DATA 0000,0000
4150 ' 横ミヤノ白
4160 DATA 80,81,82
4170 DATA 0700,00F0
4180 DATA 7700,00FF
4190 DATA 7707,FFFF
4200 DATA 7777,FFFF
4210 DATA FFFF,BB69
4220 DATA FFBF,8BBB
4230 DATA FF0B,B0B8
4240 DATA BF00,00BB
4250 DATA 0000,0000
4260 ' 横ミヤノ黒
4270 DATA 81,82,83
4280 DATA 0200,0020
4290 DATA 2207,F022
4300 DATA 2202,2022
4310 DATA 2722,2222
4320 DATA 2222,AA7A
4330 DATA 22B2,ABAA
4340 DATA 22F7,80AA
4350 DATA B200,00AB
4360 DATA 0000,0000
4370 ' おわり
4380 DATA -1

```

リスト2

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0000 01 02 0A 04 05 0A 04 02-02 02 02 01 02 0A 04 :46
7000:0010 05 04 04 02 0D 0D 02 02-01 02 0A 04 04 04 02 :52
7000:0020 02 02 02 01 09 0A 0A-09 09 09 09 09 09 09 :6E
7000:0030 01 05 0A 04 04 04 03-03 03 02 02 01 03 0A 04 :4A
7000:0040 03 03 03 03 03 03 03-01 03 0A 03 03 03 0D :45
7000:0050 03 03 03 03 01 03 0A-0B 03 0C 0C 03 03 02 :5E
7000:0060 01 03 0A 0B 03 03 0C-0C 03 03 03 03 03 0E :67
7000:0070 04 03 03 03 03 03 03-01 03 06 0A 04 04 03 :3B
7000:0080 03 03 03 03 01 02 06 0A-02 04 04 03 03 03 02 :37
7000:0090 01 02 0D 0A 02 04 04 03 03 02 01 02 0A 04 :36
7000:00A0 02 04 02 03 03 03 0D-01 0D 0D 01 02 0A 04 :10B
7000:00B0 01 0D 0D 0D 01 09 0A 09-09 09 09 09 09 09 :79
7000:00C0 01 0D 0D 0A 0D 0D 0D-02 02 02 01 02 02 0A :70
7000:00D0 02 02 02 02 02 00 00-01 00 0A 00 00 00 02 :19
7000:00E0 00 00 00 00 01 00 00-00 00 00 00 00 00 :27
7000:00F0 01 00 00 0A 00 00 00-00 00 00 01 00 00 0A :16

VSUM 2B 3E 4E 5E 40 4C 5C 67 3A 4A 5E 34 33 44 68 :97

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0200 02 00 00 00 01 00 00 0A-00 03 06 03 02 02 0D :2A
7000:0210 01 00 00 0A 00 03 06 03-00 00 00 01 00 00 0A :22
7000:0220 00 03 06 03 00 00 0D-00 01 00 0A 00 03 06 :30
7000:0230 00 00 02 00 01 03 03 0A-04 03 06 03 03 03 :2F
7000:0240 01 03 03 0A 03 03 06 03-03 03 0A 03 01 10 :0B
7000:0250 0B 06 03 03 03 0A 03-01 03 0B 03 0B 03 06 :53
7000:0260 03 03 0A 03 01 03 0B 03-0B 06 05 03 03 0A :59
7000:0270 01 03 0B 0D 0B 0D 06 03-03 03 0A 03 01 10 :0B
7000:0280 0B 0D 06 03 03 03 0A 03-01 03 03 06 03 0D :5A
7000:0290 03 03 0A 03 01 10 03 06-03 03 06 03 03 0A :4F
7000:02A0 05 05 03 02 03 03 03-01 0D 0D 0D 0D 0A :83
7000:02B0 07 07 07 03 0E 03 0A 03-01 03 03 06 03 03 :38
7000:02C0 03 03 0A 03 01 03 03 06-03 03 0E 03 03 0A :4A
7000:02D0 01 0E 03 06 03 0E 03 03-03 03 0A 03 01 05 :0E
7000:02E0 05 05 03 02 03 03 0A 03-01 0D 06 00 00 0D :38
7000:02F0 00 0A 00 01 0D 0D 0D 0D-0D 0D 0D 0D 0A :83

VSUM 32 3F 5A 44 41 31 56 71 44 33 53 6C 47 31 59 74 4B :CD

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0100 00 00 00 00 00 00 00-01 02 00 0A 02 02 02 00 :13
7000:0110 00 00 00 03 01 03 03 0A-0D 03 0E 0E 03 03 :56
7000:0120 01 03 03 0A 03 0B 03 03-03 0B 03 03 01 0D 03 :54
7000:0130 03 0B 03 03 01 0D 0D 03 0A 03 0B 03 0B :F
7000:0140 03 0B 03 03 01 0D 0D 03 0B 03 03 03 03 :71
7000:0150 01 0D 0D 0A 03 0B 03 03-03 0B 0D 0A 0F 03 :80
7000:0160 03 0B 0C 03 0C 0B 03 03-01 0D 0D 0A 0F 03 :81
7000:0170 0C 06 0F 03 01 0D 0A 0A-0F 03 0C 0C 06 0F :97
7000:0180 01 0D 0D 0A 03 03 03-03 06 03 03 01 02 02 :4F
7000:0190 04 02 03 03 06 03 03-00 04 05 05 02 03 :3C
7000:01A0 03 06 03 03 01 04 04 0A-04 02 03 06 07 :98
7000:01B0 01 04 04 0A 05 02 03 06-06 06 03 03 01 04 :4A
7000:01C0 04 02 03 06 0D 0D 0D-03 01 00 0A 03 06 :75
7000:01D0 03 03 03 03 01 00 03 0A-03 03 03 00 00 :27
7000:01E0 01 00 00 0A 00 03 06 03-02 00 03 01 00 :27
7000:01F0 00 03 06 03 02 00 00 02-01 00 00 0A 03 06 :27

VSUM 28 58 54 53 34 68 49 52 37 58 4F 66 4A 54 56 5C :F2

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0300 01 0D 0D 06 0D 0D 0D-06 0D 0A 00 01 07 07 :88
7000:0310 07 0B 0D 06 0D 0A 00-01 0D 0D 03 0D 06 :07
7000:0320 07 07 0A 00 01 0C 0D 0D-06 0D 0D 0D 0A :0E
7000:0330 01 07 07 08 0D 06 0D-0D 0D 0A 00 01 0D :06
7000:0340 0D 06 07 07 07 07 0A 00-01 0D 0D 06 07 07 :0D
7000:0350 0D 0A 00 01 03 06 06-0B 04 04 03 03 0A :63
7000:0360 01 03 03 06 03 03 03-03 03 0A 03 01 03 :06
7000:0370 03 03 03 03 03 03 0A 03-01 03 03 06 03 03 :38
7000:0380 03 0B 0A 03 03 03 0C 06-0C 03 03 03 0B :6A
7000:0390 01 03 0C 06 0C 03 03 03-03 0B 03 0C 01 :03
7000:03A0 02 03 03 03 03 03 03 0B-01 03 06 03 03 :0F
7000:03B0 0F 03 0B 0B 01 05 05 06-03 03 0F 03 0F :69
7000:03C0 01 05 0E 06 08 03 03 03-03 02 0E 02 01 :0E
7000:03D0 08 04 03 02 03 0E 0F 0E-01 05 0E 06 08 :0E
7000:03E0 02 02 0E 02 01 04 04 05-04 0E 0E 02 03 :51
7000:03F0 01 04 00 06 0D 0E 0E 03-03 03 03 01 04 :0F

VSUM 50 5F 85 5A 4C 6D 8E 67 4F 70 91 52 4D 6C 89 5D :7D

```


SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0400 04 0E 08 03 03 03 03 08-01 0F 0B 06 03 00 00 00 :58
7000:0410 08 00 00 08 01 09 03 06-03 09 09 09 09 03 08 :56
7000:0420 01 00 03 06 03 00 00 00-00 00 08 08 01 09 03 06 :30
7000:0430 03 09 09 09 09 09 09 09-01 0B 03 06 03 02 00 0F :6B
7000:0440 0F 0F 0F 0B 01 08 02 08-04 02 02 02 02 02 0B :69
7000:0450 01 08 02 08 04 04 08 00-00 03 08 01 08 02 08 :48
7000:0460 00 00 08 00 00 03 02 08-01 0B 02 07 08 0B 08 :5C
7000:0470 0B 03 02 0B 01 0B 02 03-03 04 0B 04 03 03 02 0B :55
7000:0480 01 0B 02 03 03 08 03-03 03 02 0B 01 0B 02 03 :54
7000:0490 03 03 03 03 03 03 02 08-01 0B 03 03 03 0E 02 :50
7000:04A0 03 03 02 0B 01 0B 02-02 02 02 02 02 02 02 0B :3C
7000:04B0 01 0B 0B 08 02 05 05 05-02 08 0B 08 01 0B 0B :78
7000:04C0 02 05 03 05 02 0B 0B 08-01 0B 0B 08 02 05 05 :65
7000:04D0 02 0B 0B 0B 01 0B 02 02-02 02 02 02 02 02 0B :4C
7000:04E0 01 0B 02 03 03 03 02 03-03 03 02 08 01 05 04 :3F
7000:04F0 04 04 03 03 03 03 02 0B-01 04 0D 05 0D 04 03 03 :4F

VSUM 3C 6F 5D 6A 28 6B 46 5D 1C 66 5D 6D 3A 5F 3F 85 :51

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0500 03 03 02 0B 01 04 05 05-05 04 04 03 03 02 0B :45
7000:0510 01 04 0D 05 0D 04 04 03-03 03 02 0B 01 04 04 :A4
7000:0520 05 04 04 03 03 03 02 0B-0D 00 04 05 00 00 00 08 :41
7000:0530 00 00 05 05 01 00 05 00 00-00 08 00 00 00 05 :26
7000:0540 01 00 05 04 03 00 00 04-04 00 05 01 01 05 05 :28
7000:0550 02 02 00 04 00 05 00-01 05 03 02 02 02 02 :20
7000:0560 00 00 05 03 01 05 00 00-00 02 00 00 05 05 :03 :1D
7000:0570 01 05 00 00 00 03 00 00-05 02 03 03 01 05 00 :A4 :20
7000:0580 04 04 00 00 05 02 03 03-01 05 05 05 05 00 05 :34
7000:0590 02 0B 0B 0B 01 0B 03 02-05 05 03 03 03 03 :20
7000:05A0 01 0B 03 02 03 03 03 03-03 02 03 0B 0B 03 02 :4D
7000:05B0 03 03 03 03 03 02 03 0B-0A 0B 03 02 03 03 :03 :45
7000:05C0 03 02 0B 0A 0A 0B 08 02-0F 0B 08 0B 0F 02 0C :8C :RD
7000:05D0 0A 0C 08 02 0F 0B 03 0F-0F 0B 0C 0A 0C 0B :96
7000:05E0 03 0B 03 03 02 0B 0C-0A 0C 0B 02 03 03 03 :67
7000:05F0 03 02 0B 10 0A 10 0B 02-02 02 02 02 02 0B :10 :6E

VSUM 29 41 4F 57 49 50 45 4C 5C 42 45 4C 3E 40 44 5D :88

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0600 0A 10 08 0B 02 0C 02 0A-02 0B 10 0A 10 0B 0B :A4
7000:0610 02 0C 02 0C 02 0B 04 03-03 08 10 0B 02 0E 0C :90
7000:0620 02 0B 10 0B 0A 0B 0B 10-02 0B 02 0B 02 10 0B :9A
7000:0630 0A 0D 0B 08 02 02 0B 02-08 0B 08 0A 0D 0D 0B :9B
7000:0640 02 0B 02 0B 02 0B 0D 0D-0A 02 0D 02 02 02 02 :87
7000:0650 02 0D 0D 02 0A 02 02 02-02 0D 02 02 02 02 :54
7000:0660 0A 02 0D 0D 0D 0D 02 0D-0D 0D 02 0A 02 02 :88
7000:0670 02 02 02 02 02 02 02 02-0A 02 0D 0D 0D 0D :6A
7000:0680 0D 0D 0D 0A 0A 02 0D 0D-0D 0D 0D 0D 0D 0D :C2
7000:0690 0A 02 02 02 02 02 02 02-02 02 02 02 02 02 :47
7000:06A0 0D 0D 05 02 03 03 0B-0A 02 05 0D 0D 0D 05 :89
7000:06B0 02 03 03 0B 0A 02 02 02-02 02 02 02 0C 0B :4F
7000:06C0 0A 0D 0D 02 05 05 05 02-0D 0D 0D 0A 03 03 :7E
7000:06D0 03 03 03 03 03 03 03 03-0D 03 03 03 0F 03 :5F
7000:06E0 03 03 03 02 01 03 03 0F-05 06 05 0F 03 03 :48
7000:06F0 01 03 0F 05 06 0E 06 05-0F 03 03 02 01 0F :05 :69

VSUM 5F 91 87 74 52 6B 52 89 7C 76 7F A4 76 99 7D 84 :A8

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:0700 02 0E 02 06 05 0F 03 02-05 0F 02 0E 05 0F 03 :69
7000:0710 06 05 0F 02 04 05 05 05-02 0E 02 0E 05 0F :02 :69
7000:0720 01 03 0F 05 07 0E 06 05-0F 03 03 02 01 04 :04 :67
7000:0730 05 07 05 0F 03 03 03 02-01 04 0B 04 0F 05 03 :65
7000:0740 03 03 03 02 01 04 04 04-04 0F 04 03 03 03 :02 :3D
7000:0750 0D 0D 03 03 03 03 0D 03-03 03 02 0D 00 00 :58
7000:0760 00 0D 00 00 00 00 00 00-00 00 00 00 00 00 :1A
7000:0770 00 00 00 0D 0D 0D 00 00-00 00 00 00 00 00 :27
7000:0780 0D 00 00 00 0D 00 00 00-00 00 00 00 00 00 :1A
7000:0790 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07B0 0D 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:07F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00

VSUM 38 3A 35 21 2E 43 18 21 28 2C 1D 39 33 27 24 :C1

リスト3

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1000 20 00 08 00 00 00 00 00-00 00 00 00 00 00 :5E
7000:1010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :AA
7000:1020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :DD
7000:1030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :31
7000:1040 00 00 00 00 00 00 00 00-11 11 19 99 9A 9A 7B :4E
7000:1050 8B 00 00 00 00 00 00 00-7B BB-BB 99 9A 9A 7B :22
7000:1060 8B BB 00 00 00 00 7B BB BB-BB 99 9A 9A 7B BB :98
7000:1070 8B 99 99 00 7B BB BB BB-BB 99 9A 9A 71 BB BB :C0
7000:1080 99 99 99 00 00 20 00-08 00 00 00 BB 99 9A :DD
7000:1090 9A 7B 81 BB BB 99 99 99-99 00 00 00 99 9A :37
7000:10A0 7B BB 81 BB 99 99 99 99-00 00 00 00 00 9A 7B :D5
7000:10B0 8B BB 81 99 99 99 99 99-00 00 00 00 00 00 7B :C1
7000:10C0 BB BB 91 99 99 99 00 00-00 00 00 00 00 00 BB :8D
7000:10D0 BB 99 91 99 99 00 00 00-00 00 00 00 00 00 :17
7000:10E0 99 99 91 99 00 00 00 00-00 00 00 00 00 00 :5C
7000:10F0 00 99 91 00 00 00 00 00-00 00 00 00 00 00 :2A

VSUM D9 64 91 73 9A 9A DC 72 3A 33 E7 EC 79 53 48 9B :B2

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1100 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :28
7000:1110 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :54
7000:1120 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :A8
7000:1130 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :FC
7000:1140 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :50
7000:1150 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :A4
7000:1160 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :F8
7000:1170 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :4C
7000:1180 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :A0
7000:1190 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :22
7000:11A0 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :A2
7000:11B0 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :4E
7000:11C0 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :FA
7000:11D0 00 00 00 00 00 00 00 00-AA-AA 00 00 00 00 :A6
7000:11E0 00 00 00 00 00 00 00 00-00 00 00 00 00 :52
7000:11F0 00 00 00 00 00 00 00 00-00 00 00 00 00 :54

VSUM FC FC 1C FC 04 FC A6 FA A4 A4 A4 A4 C4 A4 02 A6 :50

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1200 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :00
7000:1210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :28
7000:1220 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :35
7000:1230 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :AB
7000:1240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 :21
7000:1250 7B BB BB BB BB BB BA-00 00 00 00 00 00 7B :12
7000:1260 BB BB BB BB BB BB BB-BA 00 00 00 00 00 00 7B :BC :C8
7000:1270 BB BB BB BB BB BB BB BB-BA 00 00 00 00 7B BB :3E
7000:1280 BB BB BB BB BB BB BB BB BB-BA 99 7B BB BB :92
7000:1290 BB BB BB BB BB BB BB BB BB BB-99 00 00 20 :C2
7000:12A0 08 00 00 99 7B BB BB BB BB BB BB BB BB BB :25
7000:12B0 99 00 00 00 99 7B BB BB BB BB BB BB BB BB :99 :09
7000:12C0 00 00 00 00 99 7B BB BB BB BA 99 7B BB BB :92
7000:12D0 00 00 00 00 00 99 7B BB BB BB BB BB 99 00 :54
7000:12E0 00 00 00 00 00 00 99 7B BB BB BB 99 00 00 :DE
7000:12F0 00 00 00 00 00 00 00 99 7B BB 99 00 00 00 :68

VSUM 08 22 DD 31 EB EB EB EB 52 DF D9 1E C0 20 00 :F7

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :32
7000:1310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1320 00 00 00 00 00 00 20 08 00 00 00 00 00 00 00 00 :F4
7000:1330 CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :CC
7000:1340 CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :09
7000:1350 CD CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :83
7000:1360 CD CC DC 00 00 00 00 00 00 00 00 00 00 00 00 00 :4E
7000:1370 CD 99 9C 9C 00 00 00 00 00 00 00 00 00 00 00 00 :5B
7000:1380 9C CC CC 00 00 00 00 00 00 00 00 00 00 00 00 00 :32
7000:1390 CD CC CC 00 00 00 00 00 00 00 00 00 00 00 00 00 :05
7000:13A0 CC 99 CC CC CC CC CC 00 00 00 00 00 00 00 00 :D1
7000:13B0 CC CD D9 9C 9C 9C D9 9C 9C 9C 9C 9C 9C 9C 9C :B5
7000:13C0 CC CC CC CC 9D 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C :3F
7000:13D0 00 CC 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C :A5
7000:13E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FB
7000:13F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :D3

VSUM 9A 64 E5 94 69 02 8C 9F 62 13 8E 99 CE E5 A6 09 0B

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :2D
7000:1410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :96
7000:1420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1430 20 00 08 00 00 00 00 00 00 00 00 00 00 00 00 :80
7000:1440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :AD
7000:1450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :B6
7000:1460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :F8
7000:1470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :0F
7000:1480 CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :3A
7000:1490 9C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :8E
7000:14A0 CC 99 9C 00 00 00 00 00 00 00 00 00 00 00 00 :E9
7000:14B0 CC 94 94 9C 00 00 20 08 00 00 00 9C 00 00 00 :CD
7000:14C0 96 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :F6
7000:14D0 9C D9 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C :21
7000:14E0 DD 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C 9C :6C
7000:14F0 6D CC 94 99 CC CD 00 00 00 00 00 00 00 00 00 :CB

VSUM 99 13 F5 D0 8D 4C D6 62 5E 55 A7 C9 20 46 32 3C :79

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1500 C9 CD D9 CC 99 00 00 00 00 00 00 00 00 00 00 :04
7000:1510 C9 CD CC D9 00 00 00 00 00 00 00 00 00 00 00 :0A
7000:1520 00 C9 CD 00 00 00 00 00 00 00 00 00 00 00 00 :96
7000:1530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :28
7000:1540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :2C
7000:1550 00 00 00 00 00 00 99 93 33 33 33 33 33 33 33 :92
7000:1560 00 00 00 00 00 00 99 93 33 33 33 33 33 33 33 :F8
7000:1570 00 00 00 00 99 93 33 33 33 33 33 33 33 33 33 :0E
7000:1580 00 00 00 99 93 33 33 33 33 33 33 33 33 33 33 :C4
7000:1590 00 99 93 33 33 33 33 33 33 33 33 33 33 33 33 :00
7000:15A0 00 99 93 33 33 33 33 33 33 33 33 33 33 33 33 :90
7000:15B0 99 93 33 33 33 33 33 33 33 33 33 33 33 33 33 :F3
7000:15C0 00 00 20 08 00 00 00 33 33 33 33 33 33 33 33 :97
7000:15D0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :99
7000:15E0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :31
7000:15F0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :CB

VSUM C4 27 9A 5D 9F F8 2B 91 5E FE FE CB B8 65 3A FF :AA

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1600 33 00 00 00 00 00 00 00 00 00 00 00 33 33 33 :65
7000:1610 33 00 00 00 00 00 00 00 00 00 00 00 33 33 33 :FF
7000:1620 00 00 00 00 00 00 00 00 00 00 00 00 33 33 33 :06
7000:1630 00 00 00 00 00 00 00 00 00 00 00 00 33 33 33 :28
7000:1640 00 00 00 00 00 00 00 00 00 00 00 00 33 33 33 :00
7000:1650 00 00 00 33 33 33 33 33 33 33 33 33 33 33 :CC
7000:1660 00 00 33 33 33 33 33 33 33 33 33 33 33 33 :92
7000:1670 00 33 33 33 33 33 33 33 33 33 33 33 33 33 :38
7000:1680 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :99
7000:1690 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :97
7000:16A0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :FD
7000:16B0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :E3
7000:16C0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :99
7000:16D0 08 00 00 33 33 33 33 33 33 33 33 33 33 33 :31
7000:16E0 33 00 00 33 33 33 33 33 33 33 33 33 33 33 :9F
7000:16F0 00 00 00 00 00 33 33 33 33 33 33 33 33 33 :FE

VSUM A0 65 65 CB 64 64 31 FE BB 98 6D 65 32 98 BB 98 :CE

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1700 00 00 00 00 00 00 00 33 33 33 33 33 33 33 33 :2A
7000:1710 00 00 00 00 00 00 00 33 33 33 33 33 33 33 33 :00
7000:1720 00 00 00 00 00 00 00 00 33 33 33 33 33 33 33 :CC
7000:1730 00 00 00 00 00 00 00 00 00 33 33 33 33 33 33 :66
7000:1740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1750 00 00 00 20 08 00 00 00 00 00 00 00 00 00 :58
7000:1760 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :39
7000:1770 33 33 00 00 00 00 00 00 00 00 00 00 33 33 33 :FF
7000:1780 33 33 33 00 00 00 00 00 00 00 00 00 33 33 33 :65
7000:1790 33 33 33 33 00 00 00 00 00 00 00 00 33 33 33 :CB
7000:17A0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :31
7000:17B0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :97
7000:17C0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :FD
7000:17D0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :33
7000:17E0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :97
7000:17F0 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :64

VSUM FE CB 98 65 52 FF 07 FF 32 98 EB CB A0 65 65 CB :D2

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1800 00 33 33 33 33 33 33 33 33 33 33 33 33 33 33 :FE
7000:1810 00 00 33 33 33 33 33 33 33 33 33 33 33 33 :98
7000:1820 00 00 00 33 33 33 33 33 33 33 33 33 33 33 :02
7000:1830 00 00 00 00 33 33 33 33 33 33 33 33 33 33 :CC
7000:1840 00 00 00 00 00 33 33 33 33 33 33 33 33 33 :66
7000:1850 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1860 20 00 08 00 00 00 00 00 00 00 00 00 00 00 :79
7000:1870 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :A5
7000:1880 00 00 00 00 00 00 00 00 00 00 55 7A AA AA :74
7000:1890 00 00 00 00 00 00 00 00 00 00 55 55 55 7A :1E
7000:18A0 00 00 00 00 00 00 00 00 15 15 55 55 15 7A :A7
7000:18B0 AA 00 00 00 00 00 00 55 55 55 55 55 51 7A :AA
7000:18C0 A7 AA 00 00 00 00 55 55 51 17 55 15 57 55 :7A
7000:18D0 AA A7 AA 00 AA 95 51 55 55 57 55 55 55 55 :7A
7000:18E0 AA AA A7 AA 00 00 20 08 00 00 AA 95 55 55 :00
7000:18F0 55 55 55 55 55 55 7A AA-AA 00 00 00 AA 95 55 :B5

VSUM 1A 83 14 98 CB 3E 94 86 76 66 C0 31 59 AF 1E CB :2A

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1900 17 55 17 55 55 55 55 7A 00 00 00 00 00 00 :98
7000:1910 55 57 55 55 15 55 55 00 00 00 00 00 00 :95
7000:1920 51 51 57 71 55 55 00 00 00 00 00 00 00 :AA
7000:1930 93 55 55 55 55 55 00 00 00 00 00 00 00 :E9
7000:1940 AA 95 55 55 00 00 00 00 00 00 00 00 00 :E9
7000:1950 00 AA 95 55 00 00 00 00 00 00 00 00 00 :3F
7000:1960 00 00 00 00 00 00 00 00 00 00 00 20 08 08 :28
7000:1970 07 70 00 00 00 00 00 7B B9 00 00 00 00 00 :AB
7000:1980 76 60 00 00 00 00 7B B9 99 BA 00 00 00 00 :5D
7000:1990 00 00 00 00 00 7B B9 99 00 00 00 00 00 :6D
7000:19A0 70 00 00 7B B9 99 BB BB BB BB BA 00 00 00 :82
7000:19B0 70 00 00 7B B9 99 BB BB BB BB BB BA 00 00 :D5
7000:19C0 70 00 7B B9 99 BB BB BB BB BB BB 99 00 00 :2A
7000:19D0 70 7B B9 99 BB BB BB BB BB BB BB 99 00 :C3
7000:19E0 7B B9 99 BB BB BB BB BB BB BB BB 99 99 :2F
7000:19F0 00 00 20 08 00 00 99 BB BB BB BB BB BB BB :15

VSUM 24 95 EF 4D 5F FD 63 83 2B 53 9C FB 02 1F 4A D7 :8E

```

```

SEGM:OFFS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
7000:1A00 88 88 88 88 88 88 88 88 88 88 88 88 88 :6B
7000:1A10 98 88 88 88 88 88 88 88 88 88 88 88 88 :99
7000:1A20 88 88 88 88 88 88 88 88 88 88 88 88 88 :63
7000:1A30 88 88 88 88 88 88 88 88 88 88 88 88 88 :60
7000:1A40 78 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1A50 00 00 00 00 00 00 00 00 00 00 00 00 00 :CC
7000:1A60 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7000:1A70 00 00 00 00 00 00 00 00 20 08 00 00 00 :28
7000:1A80 00 00 07 76 78 00 00 00 00 00 00 00 00 :F8
7000:1A90 78 00 6A BD 00 00 00 00 00 00 00 00 00 :00
7000:1AA0 00 07 2B B7 AF AB 00 00 00 00 00 00 00 :43
7000:1AB0 78 B2 B7 AF FA 00 00 00 00 00 00 00 00 :3D
7000:1AC0 07 BB 22 B7 FA AB 00 00 00 00 00 00 00 :EA
7000:1AD0 78 B2 B7 FA FA 00 00 00 00 00 00 00 :00
7000:1AE0 BB B2 BB B7 AA FA AB-00 00 00 00 00 00 :75
7000:1AF0 BB 22 BB B7 AA FF FA AB-00 00 00 00 20 :DC

VSUM C2 F4 2A 36 73 42 A9 05 8B 72 F5 6C 23 F8 3E AC :DC

```

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:1800	00	00	00	00	00	07	BB	BB	BB	AA	AA	AA	AA	AA	00	:86	
7000:1810	00	00	00	00	00	7B	BB	BB	BB	AA	AA	AA	AA	AA	00	:87	
7000:1820	00	00	00	00	00	00	BB	BB	BB	BB	BB	BB	BB	BB	AA	:00	
7000:1830	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:8F	
7000:1840	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:2A	
7000:1850	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:C5	
7000:1860	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:60	
7000:1870	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:90	
7000:1880	00	00	00	00	00	20	08	00	00	00	00	00	00	00	00	:77	
7000:1890	88	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:77	
7000:18A0	AA	88	00	00	00	00	00	00	00	00	00	00	00	00	00	:77	
7000:18B0	AA	88	88	00	00	00	00	00	00	00	00	00	00	00	00	:77	
7000:18C0	AA	AA	88	88	00	00	00	00	00	00	00	00	00	00	00	:77	
7000:18D0	AA	AA	88	88	88	00	00	00	00	00	00	00	00	00	00	:77	
7000:18E0	AA	AA	88	88	88	90	00	00	00	00	00	00	00	00	00	:77	
7000:18F0	AA	AA	88	88	90	00	00	00	00	00	00	00	00	00	00	:77	

VSUM	8A	B6	EA	20	3A	51	FA	EE	C9	01	AF	4C	4D	FD	53	04	:1D
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:1C00	AA	AA	AA	88	88	90	00	00	00	00	20	08	00	00	00	:86	
7000:1C10	00	97	BB	BB	BB	BB	AA	AA	AA	AA	80	90	00	00	00	:3B	
7000:1C20	00	99	BB	BB	BB	BB	AA	AA	AA	AA	90	00	00	00	00	:5F	
7000:1C30	00	99	BB	BB	BB	BB	AA	AA	AA	AA	99	00	00	00	00	:C6	
7000:1C40	00	00	99	99	BB	BB	AA	AA	AA	99	00	00	00	00	00	:40	
7000:1C50	00	00	00	00	99	99	BB	BB	BB	AA	99	00	00	00	00	:C9	
7000:1C60	00	00	00	00	99	99	BB	BB	BB	BB	00	00	00	00	00	:64	
7000:1C70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1C80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1C90	20	08	00	00	00	00	00	00	00	00	7B	BA	00	00	00	:5D	
7000:1CA0	00	00	00	00	00	00	00	00	00	00	7B	BB	BB	BA	00	:AB	
7000:1CB0	00	00	00	00	00	00	00	00	00	00	7B	BB	BB	BB	BA	:00	
7000:1CC0	00	00	00	00	00	00	00	00	00	00	00	7B	BB	BB	BB	:97	
7000:1CD0	00	00	00	00	00	00	00	00	00	00	7B	BB	BB	BB	BB	:53	
7000:1CE0	BA	00	00	00	00	00	07	BB	BB	BB	BB	BB	BB	BB	BB	:53	
7000:1CF0	BB	00	00	00	00	00	79	97	BB	BB	BB	BB	BB	BB	BB	:53	

VSUM	3F	25	7C	0D	A6	D0	6B	AT	EC	FE	FB	FD	24	61	A6	EB	:6D
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:1D00	BB	88	00	00	00	07	9A	00	97	BB	BB	BB	BB	BB	BB	:3E	
7000:1D10	88	88	00	00	00	00	20	00	00	00	00	9A	AA	AA	97	:7E	
7000:1D20	BB	BB	BB	BB	BB	88	88	88	88	99	00	00	00	00	00	:9A	
7000:1D30	97	BB	BB	BB	BB	88	88	88	88	99	90	00	00	00	00	:9A	
7000:1D40	AA	7B	BB	88	88	99	90	00	00	00	00	00	00	00	00	:9A	
7000:1D50	AA	7B	BB	88	88	99	90	00	00	00	00	00	00	00	00	:9A	
7000:1D60	AA	7B	BB	88	88	99	90	00	00	00	00	00	00	00	00	:06	
7000:1D70	09	7B	BB	88	88	90	00	00	00	00	00	00	00	00	00	:CC	
7000:1D80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1D90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:28	
7000:1DA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:EE	
7000:1DB0	00	00	00	00	00	00	77	70	07	77	00	00	00	00	00	:65	
7000:1DC0	00	00	00	00	00	00	77	70	0F	00	07	88	00	00	00	:75	
7000:1DD0	00	00	00	00	00	00	77	77	F2	2F	77	88	00	00	00	:95	
7000:1DE0	00	00	00	00	00	00	77	77	72	72	7A	88	88	88	00	:E7	
7000:1DF0	00	00	00	00	00	00	77	7B	BB	BB	7A	AA	88	88	88	:67	

VSUM	9A	85	06	EC	68	32	53	E4	5D	54	FB	53	8E	A8	C4	56	:37
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:1E00	00	00	00	00	00	07	BB	BB	BB	BB	AA	AA	88	88	90	00	:DD
7000:1E10	00	00	00	00	00	77	BB	BB	BB	BB	AA	AA	88	88	90	00	:01
7000:1E20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:8F
7000:1E30	AA	AA	80	90	00	00	00	00	00	00	9B	BB	BB	BB	AA	:3F	
7000:1E40	AA	AA	80	90	00	00	00	00	00	00	9B	BB	BB	BB	AA	:5D	
7000:1E50	AA	AA	99	00	00	00	00	00	00	00	9B	BB	BB	BB	AA	:EB	
7000:1E60	AA	99	00	00	00	00	00	00	00	00	00	00	00	00	00	:72	
7000:1E70	99	00	00	00	00	00	00	00	00	00	00	99	99	99	AA	:FD	
7000:1E80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1E90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1EA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1EB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1EC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1ED0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
7000:1EE0	07	BB	AA	AA	AA	AA	BB	00	00	00	00	00	00	00	00	:2D	
7000:1EF0	99	99	AA	AA	AA	AA	BB	00	00	00	00	00	00	00	00	:2B	

VSUM	00	FB	E8	5D	09	7C	DC	F1	74	1F	26	95	2E	60	EB	5C	:B5
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:1F00	99	AA	AA	AA	AA	AA	AA	BB	BB	BB	00	00	00	00	00	00	:BD
7000:1F10	99	9A	9A	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	00	00	:8C
7000:1F20	99	9A	AA	AA	AA	AA	BB	BB	BB	BB	00	00	00	00	00	00	:7C
7000:1F30	08	00	00	00	00	07	B9	99	99	99	AA	AA	BB	BB	BB	90	:24
7000:1F40	00	00	00	00	00	0A	7B	99	99	99	BB	BB	BB	BB	BB	BB	:05
7000:1F50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:8C
7000:1F60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:DD
7000:1F70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:2D
7000:1F80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:5B
7000:1F90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
7000:1FA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
7000:1FB0	00	00	00	00	00	20	08	00	00	00	00	00	00	00	00	00	:A3
7000:1FC0	BB	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:7B
7000:1FD0	AB	BB	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:A4
7000:1FE0	AA	AB	BB	00	00	00	00	00	00	00	00	00	00	00	00	00	:D9
7000:1FF0	AA	AA	AB	BB	00	00	00	00	00	00	00	00	00	00	00	00	:CA

VSUM	8D	EE	54	B9	2F	ED	BF	5C	3B	26	97	29	87	65	7B	35	:AC
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

SEGM:OFFS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	SUM
7000:2000	AA	7A	DB	00	00	00	00	00	00	00	BB	77	7B	9A	AA	:BC	
7000:2010	AA	AA	AB	7B	88	88	00	00	00	00	BB	77	7B	9A	AA	:32	
7000:2020	AA	AA	BB	7B													


```

1000 ' AZON-VA
1010 '
1020 '
1030 '
1040 '
1050 KEY 1,"WIDTH 80,25"+CHR$(13)
1060 KEY 2,"SCREEN ,2"+CHR$(13)
1070 '=====
1080 CLEAR ,&H7000: DEFINT A-Z
1090 DEVICE=& DEVICE=13とすると、ジョイスティックで遊べるよ！
1100 PALETTE MODE 2,2
1110 DEF FNCG(R,B)=G*64+R*8+B
1120 PALETTE
1130 PALETTE 3,FCN(0,2,0),1
1140 PALETTE 11,FCN(0,0,0),1
1150 PALETTE 19,FCN(3,5,0),1
1160 PALETTE 24,FCN(0,0,0),1
1170 PALETTE 25,FCN(1,1,1),1
1180 PALETTE 26,FCN(3,3,3),1
1190 PALETTE 27,FCN(5,5,5),1
1200 PALETTE 28,FCN(2,0,0),1
1210 PALETTE 29,FCN(4,0,0),1
1220 PALETTE 30,FCN(0,2,0),1
1230 PALETTE 31,FCN(0,4,0),1
1240 CONSOLE 0:"WIDTH 40,25: SCREEN ,0
1250 SPRITE SCREEN 1
1260 MOUSE 0: MOUSE 1,0,0,0: MOUSE 3,0,20: MOUSE 3,1,20: MOUSE 4,16,16,420,368
1270 SCREEN WIDTH 4,320,4 '80:320*200*16c
1280 SPAN=384*7
1290 BUFFER 0,SPAN,192,4
1300 SCREEN WINDOWS 0,(0,0),200,0
1310 BUFFER=0
1320 'OUT &H100,INP(&H100) OR &H40 '走査線間隔がなくなるよ
1330 '
1340 '地上パターンデータをロード
1350 WX=32: WY=8: S=4
1360 NB=(WX*5+15)*16*WY*2+4: GB=NB*2+1
1370 ALLNEED=GB*2+2: BSLN=ALLNEED*2
1380 DIM GGX(ALLNEED)
1390 SGM=VARPTR(GGX(0),1): OFS=VARPTR(GGX(0),0)
1400 DEF SEG=SGM: BLOAD "GROUND.PAT",OFS
1410 '
1420 'スプライトパターン設定
1430 LOAD PATTERN "GAME.SPR",ALL
1440 DEF SPRITE 1 AS 1
1450 DEF SPRITE 7 AS 10
1460 DEF SPRITE 8 AS 11
1470 DEF SPRITE 2 AS 12
1480 DEF SPRITE 3 AS 12
1490 DEF SPRITE 4 AS 12
1500 DEF SPRITE 5 AS 12
1510 DEF SPRITE 6 AS 13
1520 '
1530 'クリップ割り込み
1540 CLIP WINDOW(0,0)-(-639,399)
1550 ON CLIP(2) GOSUB *M1.OFF
1560 ON CLIP(3) GOSUB *M2.OFF
1570 ON CLIP(4) GOSUB *M3.OFF
1580 ON CLIP(5) GOSUB *M4.OFF
1590 ON CLIP(6) GOSUB *GM.EXP
1600 ON CLIP(9) GOSUB *B1.OFF
1610 ON CLIP(18) GOSUB *B2.OFF
1620 ON CLIP(11) GOSUB *B3.OFF
1630 ON CLIP(12) GOSUB *B4.OFF
1640 ON CLIP(13) GOSUB *B5.OFF
1650 ON CLIP(14) GOSUB *B6.OFF
1660 ON CLIP(15) GOSUB *B7.OFF
1670 ON CLIP(16) GOSUB *B8.OFF
1680 ON CLIP(17) GOSUB *B9.OFF
1690 ON CLIP(18) GOSUB *B10.OFF
1700 ON CLIP(19) GOSUB *B11.OFF
1710 ON CLIP(20) GOSUB *B12.OFF
1720 ON CLIP(21) GOSUB *T1.OFF
1730 ON CLIP(22) GOSUB *T2.OFF
1740 ON CLIP(23) GOSUB *T3.OFF
1750 ON CLIP(24) GOSUB *T4.OFF
1760 ON CLIP(25) GOSUB *T5.OFF
1770 ON CLIP(26) GOSUB *T6.OFF
1780 ON CLIP(27) GOSUB *T7.OFF
1790 ON CLIP(28) GOSUB *T8.OFF
1800 ON CLIP(29) GOSUB *T9.OFF
1810 ON CLIP(30) GOSUB *T10.OFF
1820 CLIP ON
1830 GOTO *INT4
1840 '
1850 *M1.OFF: TURN OFF SPRITE 2: STOP SPRITE 2: RETURN
1860 *M2.OFF: TURN OFF SPRITE 3: STOP SPRITE 3: RETURN
1870 *M3.OFF: TURN OFF SPRITE 4: STOP SPRITE 4: RETURN
1880 *M4.OFF: TURN OFF SPRITE 5: STOP SPRITE 5: RETURN
1890 *B1.OFF: TURN OFF SPRITE 9: STOP SPRITE 9: RETURN
1900 *B2.OFF: TURN OFF SPRITE 10: STOP SPRITE 10: RETURN
1910 *B3.OFF: TURN OFF SPRITE 11: STOP SPRITE 11: RETURN
1920 *B4.OFF: TURN OFF SPRITE 12: STOP SPRITE 12: RETURN
1930 *B5.OFF: TURN OFF SPRITE 13: STOP SPRITE 13: RETURN
1940 *B6.OFF: TURN OFF SPRITE 14: STOP SPRITE 14: RETURN
1950 *B7.OFF: TURN OFF SPRITE 15: STOP SPRITE 15: RETURN
1960 *B8.OFF: TURN OFF SPRITE 16: STOP SPRITE 16: RETURN
1970 *B9.OFF: TURN OFF SPRITE 17: STOP SPRITE 17: RETURN
1980 *B10.OFF: TURN OFF SPRITE 18: STOP SPRITE 18: RETURN
1990 *B11.OFF: TURN OFF SPRITE 19: STOP SPRITE 19: RETURN
2000 *B12.OFF: TURN OFF SPRITE 20: STOP SPRITE 20: RETURN
2010 *T1.OFF

```

```

2020 TURN OFF SPRITE 21: STOP SPRITE 21: XING(21)=0: LIFE(21)=0: RETURN
2030 *T2.OFF
2040 TURN OFF SPRITE 22: STOP SPRITE 22: XING(22)=0: LIFE(22)=0: RETURN
2050 *T3.OFF
2060 TURN OFF SPRITE 23: STOP SPRITE 23: XING(23)=0: LIFE(23)=0: RETURN
2070 *T4.OFF
2080 TURN OFF SPRITE 24: STOP SPRITE 24: XING(24)=0: LIFE(24)=0: RETURN
2090 *T5.OFF
2100 TURN OFF SPRITE 25: STOP SPRITE 25: XING(25)=0: LIFE(25)=0: RETURN
2110 *T6.OFF
2120 TURN OFF SPRITE 26: STOP SPRITE 26: XING(26)=0: LIFE(26)=0: RETURN
2130 *T7.OFF
2140 TURN OFF SPRITE 27: STOP SPRITE 27: XING(27)=0: LIFE(27)=0: RETURN
2150 *T8.OFF
2160 TURN OFF SPRITE 28: STOP SPRITE 28: XING(28)=0: LIFE(28)=0: RETURN
2170 *T9.OFF
2180 TURN OFF SPRITE 29: STOP SPRITE 29: XING(29)=0: LIFE(29)=0: RETURN
2190 *T10.OFF
2200 TURN OFF SPRITE 30: STOP SPRITE 30: XING(30)=0: LIFE(30)=0: RETURN
2210
2220 *衝突
2230 IF COLLISION(0)>8 THEN RETURN
2240 C.C=COLLISION(2): IF 20<C.C AND C.C<31 THEN AM.C=2: GOSUB *HITTING
2250 C.C=COLLISION(3): IF 20<C.C AND C.C<31 THEN AM.C=3: GOSUB *HITTING
2260 C.C=COLLISION(4): IF 20<C.C AND C.C<31 THEN AM.C=4: GOSUB *HITTING
2270 C.C=COLLISION(5): IF 20<C.C AND C.C<31 THEN AM.C=5: GOSUB *HITTING
2280 C.C=COLLISION(1): IF C.C>8 THEN *BREAK
2290 RETURN
2300
2310 *HITTING
2320 IF XING(C.C) THEN RETURN
2330 BEEP 1
2340 XING(C.C)=1: 爆発中フラグオン
2350 STOP SPRITE AM.C: TURN OFF SPRITE AM.C: ミサイル消去
2360 DEF SPRITE C.C AS 20: TURN ON SPRITE C.C: 爆発設定
2370 SPRITE C.C,"S40U4R4L4D4H800.800",1: START SPRITE C.C
2380 BEEP 0
2390 SCORE=SCORE+1: GOSUB *DSP.SCORE
2400 BONUS=BONUS+1: IF BONUS=100 THEN BONUS=0: PLAYER=PLAYER+1: LOCATE 39-PLAYER
2410 PRINT " ": PLAY "07 T250 L16 07 CCC"
2420
2430 *BREAK
2440 IF XING(C.C) THEN RETURN
2450 VOICE REG 6,20: VOICE REG 7,&HC7: PLAY #2,...,"SIT100L32 V15CC V12CC V10CC
V8CCCC","V15C","V15C"
2460 DEF SPRITE 1 AS 20: SPRITE 1,"S500L20R20L10U10D30L20R20L10": START SPRITE
2470 FOR T=0 TO 10000: NEXT: TURN OFF SPRITE 1
2480 FOR T=0 TO 10000: NEXT: STOP SPRITE: TURN OFF SPRITE
2490 DEF SPRITE 1 AS 1
2500 PLAYER=PLAYER-1: IF PLAYER>0 THEN *MAIN.CONT
2510 *ゲームオーバー
2520 LOCATE 15,15: PRINT "GAME OVER": RETURN *MAIN
2530
2540 *INT4
2550 DIM XING(30)
2560 DIM SPR(30)
2570 DIM LIFE(30)
2580
2590 OUT &H44,7: OUT &H45,INP(&H45) AND &H3F
2600 CLS
2610
2620 *メインルーチンなの!
2630 *=====
2640 *MAIN
2650 DEF SEG=&H7000
2660 GOSUB *DRAW.MAP
2670 SCORE=0: ROUND=1: PLAYER=3: BONUS=0
2680
2690 LOCATE 0,15: PRINT TAB(39): LOCATE 17,15: PRINT "DEMO": GOSUB *DEMO
2700 *MAIN.CONT
2710 DIST=0: X=230: Y=204
2720 ROLL TO (0,0): ROLL ,-192
2730 TURN ON SPRITE 1: TURN ON SPRITE 7
2740 MOUSE 1,200,200,0
2750 CLS: PRINT "SCORE": GOSUB *DSP.SCORE
2760 LOCATE 0,1: PRINT "ROUND":ROUND:
2770 FOR T=1 TO PLAYER-1: LOCATE 38-T,24: PRINT " ": NEXT
2780 *-----
2790 *MAIN.START
2800 RESTORE *STAGE1: READ TIMING: DIST=0
2810 LOCATE SPRITE 1,X,Y: LOCATE SPRITE 7,X+100,Y
2820
2830 WHILE DIST/2<SPAN
2840 DELAY=100
2850 ROLL -1,1
2860 GOSUB *MOVE
2870 IF COLLISION(-1) THEN GOSUB *衝突
2880 FOR T=0 TO DELAY: NEXT
2890 GOSUB *MOVE
2900
2910 DELAY=100
2920 ROLL 0,1
2930 IF COLLISION(-1) THEN GOSUB *衝突
2940 GOSUB *MOVE
2950 DIST=DIST+4
2960 IF DIST/8>TIMING THEN GOSUB *SET.CHR: READ TIMING: DELAY=0
2970 FOR T=0 TO DELAY: NEXT
2980 VEND
2990
3000 *ROUND
3010 IF G.ENEMIES<0 THEN *完了

```

```

3828 ROUND=ROUND+1: LOCATE 5,1: PRINT ROUND:
3838 GOTO *MAIN.START
3848
3858 *DSP.SCORE
3868 LOCATE 5,0: PRINT SCORE:CHR$(20):"000": RETURN
3878 *E7
3888 LOCATE 18,10: PRINT "MISSION COMPLETE !!!"
3898 LINE INPUT WAIT 50,AS
3108 GOTO *MAIN
3118
3128 *敵キャラクター出現*****
3138 *SET.CHR
3148 READ SET.TYPE
3158 IF SET.TYPE=0 THEN *PRE.SET
3168 ON ABS(SET.TYPE) GOTO *A.MIS1,*A.MIS2,*G.MIS1,*G.MIS2
3178 BEEP: END
3188
3198
3208 *PRE.SET
3218 READ NUM
3228 N=21: WHILE N<30: IF SPRITE(N,1) THEN N=N+1: WEND: RETURN'空き探し
3238 SPR(N)=NUM: LIFE(NUM)=N
3248 LOCATE SPRITE N,LX,LY
3258 ON SET.TYPE GOSUB *PRE1,*PRE2,*PRE3,*PRE4,*PRE5,*PRE6,*PRE7,*PRE8,*PRE9,*
PRE10,*PRE11
3268 TURN ON SPRITE N: START SPRITE N
3278 RETURN
3288
3298 *PRE1: DEF SPRITE N AS 30,31,32
3308 IF RND(1)>.5
THEN LOCATE SPRITE N,500,0: SPRITE N,"C55250M-640,320": RETURN
ELSE LOCATE SPRITE N,630,100: SPRITE N,"C55300M-640,320": RETURN
3318
3328 *PRE2: DEF SPRITE N AS 30,31,32
3338 IF SPRITE(1,4)<200
THEN LOCATE SPRITE N,630,200: SPRITE N,"C55300M-640,-320": RETURN
ELSE LOCATE SPRITE N,200,0: SPRITE N,"C55200M640,320": RETURN
3348
3358 *PRE3: DEF SPRITE N AS 30,31,32
3368 IF RND(1)>.5
THEN LOCATE SPRITE N,600,0: SPRITE N,"C55300D300M-640,-320": RETURN
ELSE LOCATE SPRITE N,630,50: SPRITE N,"C55300L450M640,320": RETURN
3378
3388 *PRE4: DEF SPRITE N AS 40
3398 IF RND(1)>.5
THEN LOCATE SPRITE N,0,300: SPRITE N,"S200M100,-50U200R640": RETURN
ELSE LOCATE SPRITE N,50,300: SPRITE N,"S200M100,-50R300U400": RETURN
3408
3418 *PRE5: DEF SPRITE N AS 50
3428 IF SPRITE(1,4)>200
THEN LOCATE SPRITE N,320,0: SPRITE N,"S200G,200G320,300G650,0": RETURN
ELSE LOCATE SPRITE N,620,200: SPRITE N,"S200G320,300G,300G650,0": RETURN
3438
3448 *PRE6: DEF SPRITE N AS 60,61
3458 IF RND(1)>.5
THEN LOCATE SPRITE N,500,0: SPRITE N,"S250C0N2L200N1D150N2L200N1D150N2L2
0N1D150": RETURN
ELSE LOCATE SPRITE N,620,50: SPRITE N,"S250C0N2L200N1D150N2L200N1D150N2L2
0N1D150": RETURN
3468
3478 *PRE7: DEF SPRITE N AS 40
3488 LOCATE SPRITE N,620,0: SPRITE N,"S200T1T1T1T1G-50,350": RETURN
3498
3508 *PRE8: DEF SPRITE N AS 50
3518 IF SPRITE(1,3)>300
THEN LOCATE SPRITE N,620,200
ELSE LOCATE SPRITE N,320,0
SPRITE N,"C55400T1S200G500,50G-50,350": RETURN
3528
3538
3548 *PRE9: DEF SPRITE N AS 50
3558 IF RND(1)>.5
THEN LOCATE SPRITE N,550,0: SPRITE N,"S400M-300,150P500M400,-200": RETU
N
ELSE LOCATE SPRITE N,620,100: SPRITE N,"S200M-300,150P500M400,-200": RET
URN
3568
3578 *PRE10: DEF SPRITE N AS 40
3588 IF STAGE<3 THEN LIFE(NUM)=0: RETURN 3270
3598 IF RND(1)>.5
THEN LOCATE SPRITE N,INT(RND(1)*600)+300
ELSE LOCATE SPRITE N,0,INT(RND(1)*300)+50
SPRITE N,"S200T1T1T1T1G-40,400": RETURN
3608
3618
3628 *PRE11: DEF SPRITE N AS 70
3638 IF RND(1)>.5
THEN LOCATE SPRITE N,500,0: SPRITE N,"S200M-640,320": RETURN
ELSE LOCATE SPRITE N,630,100: SPRITE N,"S350M-640,320": RETURN
SPRITE N,"S350H630,100M-640,320": RETURN
3648
3658
3668 *空中兵器からのミサイル発射
3678 *A.MIS1
3688 READ NUM: IF LIFE(NUM)=0 THEN RETURN
3698 N=9: WHILE N<21: IF SPRITE(N,1) THEN N=N+1: WEND: RETURN'空き探し
3708 NN=LIFE(NUM): FX=SPRITE(NN,3): FY=SPRITE(NN,4)
3718 DEF SPRITE N AS 80: LOCATE SPRITE N,FX,FY
3728 XI=SPRITE(1,3): YI=SPRITE(1,4)
3738 TX=3*(XI-FX): TY=3*(YI-FY)
3748 SPRITE N,"S150D20M+TX,*TY:H-50,0":TURN ON SPRITE N: START SPRITE N
3758 RETURN
3768 *A.MIS2
3778 READ NUM: IF LIFE(NUM)=0 THEN RETURN
3788 N=9: WHILE N<21: IF SPRITE(N,1) THEN N=N+1: WEND: RETURN'空き探し
3798 NN=LIFE(NUM): FX=SPRITE(NN,3): FY=SPRITE(NN,4)
3808 DEF SPRITE N AS 81: LOCATE SPRITE N,FX,FY
3818 SPRITE N,"S150D20T1T1T1R-50,0": TURN ON SPRITE N: START SPRITE N
3828 RETURN
3838
3848 *地上兵器からのミサイル発射
3858 *G.MIS1
3868 READ QMX,GMV: 発射する地上兵器のマップ座標
3878 IF PEEK(QMX*12+GMV) AND 128 THEN RETURN
3888 N=9: WHILE N<21: IF SPRITE(N,1) THEN N=N+1: WEND: RETURN'空き探し
3898 FX=QMX*32+GMV*32+400-DIST: FY=GMV*16-QMX*16+DIST*2-4
3908 DEF SPRITE N AS 80: LOCATE SPRITE N,FX,FY
3918 XI=SPRITE(1,3): YI=SPRITE(1,4)
3928 TX=3*(XI-FX): TY=3*(YI-FY)+20: S=100+INT(RND(1)*150)
3938 SPRITE N,"S+S1U20M+TX,*TY:H-50,0":TURN ON SPRITE N: START SPRITE N

```



```

3940 RETURN
3950 *G.MIS2
3960 READ GMX, GMY: 発射する地上兵器のマップ座標
3970 IF PEEK(GMX*12+GY) AND 128 THEN RETURN
3980 N=9: WHILE N<21: IF SPRITE(N,1) THEN N=N+1: WEND: RETURN'空き探し
3990 FX=GMX*32+GMY*32+408-DIST: FY=GMY*16-GMX*16+DIST/2-4
4000 DEF SPRITE N AS 81: LOCATE SPRITE N,FX,FY
4010 SPRITE N,"S150U20TITITH-50,0": TURN ON SPRITE N: START SPRITE N
4020 RETURN
4030 'デバイス入力、機体移動
4040 '=====
4050 *MOVE
4060 IF MOUSE(2,1) THEN REP=0 ELSE IF REP=0 THEN REP=1: GOSUB *AM.FIRE: DELAY=0
4070 IF MOUSE(2,2) THEN GOSUB *GM.FIRE: DELAY=0
4080 *MOVE2
4090 IF DEVICE=0 THEN X=MOUSE(0): Y=MOUSE(1): GOTO 4130
4100 'ジョイスティック入力
4110 OUT $B44,AHE: D=255-INP($H45): IF D=0 THEN RETURN
4120 ON D GOSUB *J1,*J2,*J3,*J4,*J5,*J6,*J7,*J8,*J9,*J10
4130 LOCATE SPRITE 1,X,Y: LOCATE SPRITE 7,X*180,Y
4140 RETURN
4150 *J1: IF Y<16 THEN RETURN ELSE Y=Y+10: RETURN
4160 *J2: IF Y>360 THEN RETURN ELSE Y=Y+10: RETURN
4170 *J3
4180 *J4: IF X<16 THEN RETURN ELSE X=X+10: RETURN
4190 *J5: IF X<16 THEN *J1 ELSE IF Y<16 THEN *J4
4200 X=X-8: Y=Y-4: RETURN
4210 *J6: IF X<16 THEN *J2 ELSE IF Y>360 THEN *J4
4220 X=X-8: Y=Y+4: RETURN
4230 *J7
4240 *J8: IF X>430 THEN RETURN ELSE X=X+10: RETURN
4250 *J9: IF X>430 THEN *J1 ELSE IF Y<16 THEN *J8
4260 X=X+8: Y=Y-4: RETURN
4270 *J10: IF X>430 THEN *J2 ELSE IF Y>360 THEN *J8
4280 X=X+8: Y=Y+4: RETURN
4290 '
4300 '対空ミサイル発射
4310 *AM.FIRE
4320 IF SPRITE(2,2)=0 THEN AM=2: GOTO *FIRE
4330 IF SPRITE(3,2)=0 THEN AM=3: GOTO *FIRE
4340 IF SPRITE(4,2)=0 THEN AM=4: GOTO *FIRE
4350 IF SPRITE(5,2)=0 THEN AM=5: GOTO *FIRE
4360 RETURN
4370 *FIRE
4380 TURN ON SPRITE AM: LOCATE SPRITE AM,X+64,Y-16:SPRITE AM,"S800M640,-320":ST
ART SPRITE AM:RETURN
4390 '対地ミサイル発射
4400 *GM.FIRE
4410 IF SPRITE(6,2) THEN RETURN
4420 GMY,OBJ=INT((Y+Y/32+X/64)+3: IF GMY.OBJ<0 OR 11<GMY.OBJ THEN RETURN
4430 GMX,OBJ=INT((X+DIST*2-2051/64-Y/32)
4440 TURN ON SPRITE 6: LOCATE SPRITE 6,X+64,Y+10: SPRITE 6,"S500M60,-20M50.10H8
00,800":START SPRITE 6
4450 TURN ON SPRITE 8: LOCATE SPRITE 8,X+180,Y: SPRITE 8,"S500M-100,50":START SP
RITE 8
4460 RETURN
4470 '
4480 '対地ミサイル着弾
4490 *GM.EXP
4500 TURN OFF SPRITE 6: STOP SPRITE 6
4510 TURN OFF SPRITE 8: STOP SPRITE 8
4520 IF GMX,OBJ<0 THEN RETURN
4530 GTYPE=PEEK(GMX,OBJ*12+GMY,OBJ)
4540 IF GTYPE<11 OR (GTYPE AND 128) THEN RETURN
4550 VOICE REG 6,8: VOICE REG 7,8HC7: PLAY #2,...,"S0T100L64 V15D R64 L30 V13C V
12C V10C V8C"
4560 GPX=(GMX,OBJ+GMY,OBJ)*16
4570 GPY=((GMY,OBJ+23-(GMX,OBJ MOD 24)) MOD 24)*8
4580 PUT (GPX,GPY),GGX(GB*34),PSET,...
4590 IF GPY<180 THEN GPY=GPY+8 ELSE GPY=0
4600 PUT (GPX,GPY),GGX(GB*35),PSET,...
4610 G.ENEMIES=G.ENEMIES-1
4620 POKE (GMX,OBJ*12+GMY,OBJ),GTYPE OR 128
4630 SCORE=SCORE+1: GOSUB *DSP.SCORE
4640 DELAY=0: RETURN
4650 '
4660 '地上マップを描く
4670 *DRAW.MAP
4680 DEF SEG=$17800: BLOAD "STAGE1.MAP"
4690 ROLL TO (0,0): ROLL ,-100: LINE (0,0),-100,12,BF
4700 GXE=24*7-13: G.ENEMIES=0
4710 FOR GX=0 TO GXE: FOR GY=0 TO 11
4720 D=PEEK(GX*12+GY): IF D=0 THEN 4780 ELSE D=D-1
4730 IF D<0 AND D<16 THEN ENEMIES=G.ENEMIES+1
4740 X=(GX+GY)*16: Y=((GY+23-(GX MOD 24)) MOD 24)*8
4750 PUT (X,Y),GGX(GB*D*2),PSET,...
4760 Y=Y+8: IF Y>191 THEN Y=0
4770 PUT (X,Y),GGX(GB*D*2+GB),PSET,...
4780 NEXT: ROLL -8,16: NEXT
4790 RETURN
4800 '
4810 *DEMO
4820 ROLL TO (0,0): ROLL ,-192
4830 LOCATE SPRITE 1,230,204: TURN ON SPRITE 1
4840 *DEMO1
4850 RESTORE *STAGE1: READ TIMING: DIST=0
4860 WHILE DIST<5SPAN
4870 IF MOUSE(2,1) THEN STOP SPRITE: TURN OFF SPRITE: RETURN
4880 IF A=0 THEN A=1: ROLL -1,1 ELSE A=0: ROLL 0,1
4890 DIST=DIST+2
4900 IF DIST/5>TIMING THEN GOSUB *SET.CHR: READ TIMING: GOTO 4920
4910 FOR T=0 TO 200: NEXT
4920 WEND: GOTO *DEMO1

```

```

4930
4940 シナリオデータ
4950 A
4960 ASTAGE1
4970 DATA 20,3,1
4980 DATA 21,3,2
4990 DATA 30,3,3
5000 DATA 31,3,4
5010 DATA 35,1,5,39,-1,5,40,11,6,42,-1,5,43,-1,6,44,-1,5,45,-1,6
5020 DATA 50,3,7
5030 DATA 51,3,8
5040 DATA 52,3,9
5050 DATA 53,3,10
5060 DATA 55,1,0,56,2,0,57,1,0,58,2,0,59,1,0,60,2,0,61,1,0,62,2,0
5070
5080 DATA 70,4,12,72,4,13,74,4,14,76,4,15
5090
5100 DATA 80,2,1
5110 DATA 85,2,2
5120 DATA 90,2,3
5130 DATA 95,2,4
5140 DATA 100,-3,23,6
5150 DATA 105,-3,23,6
5160 DATA 110,-3,23,6
5170 DATA 120,6,1,122,6,2,124,6,3
5180 DATA 126,6,1,128,6,2,130,6,3
5190
5200 DATA 140,-3,30,4
5210 DATA 141,-3,31,4
5220 DATA 145,-3,30,10
5230 DATA 146,-3,31,10
5240 DATA 150,-3,30,4
5250 DATA 151,-3,31,4
5260 DATA 155,-3,30,10
5270 DATA 156,-3,31,10
5280 DATA 160,-3,30,4
5290 DATA 161,-3,31,4
5300 DATA 165,-3,30,10
5310 DATA 166,-3,31,10
5320
5330 DATA 170,10,1,172,10,2,174,-1,1,176,-1,2,178,-1,1,178,-1,2
5340
5350 DATA 180,8,1,181,5,0,182,6,0,183,6,0,184,6,0,185,6,0,186,6,0,187,6,0
5360 DATA 192,0,2,195,9,3
5370 DATA 185,-1,1,187,-1,1,189,-1,1,190,-2,2,192,-2,3,194,-2,2
5380
5390 DATA 220,6,0,223,6,0,225,6,0,230,6,0
5400 DATA 236,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5410 DATA 240,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5420 DATA 245,5,0,-0,4,0
5430 DATA 250,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5440 DATA 260,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5450 DATA 270,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5460 DATA -0,-4,60,1,-0,-4,61,2
5470 DATA 280,-3,60,5,-0,-3,59,6,-0,-3,58,7,-0,-3,57,8,-0,-3,56,9
5480
5490 DATA 290,2,0,298,3,0,291,2,0,291,3,0,292,2,0,292,3,0,296,4,0
5500 DATA 300,7,0,302,7,0,304,7,0,306,7,0
5510 DATA 307,11,1,310,11,2,314,-1,1,315,11,3,0,-1,2
5520
5530 DATA 316,-3,78,6,317,-3,78,8,318,-3,79,6,319,-3,79,8
5540 DATA 324,-3,80,2,325,-3,81,1,326,-3,81,3,327,-3,82,2,327,-4,81,2
5550 DATA 332,-3,83,5,0,-3,84,5,0,-3,82,6,0,-3,80,10
5560 DATA 340,-3,85,5,341,-3,81,11,342,-3,85,6,343,-3,82,10
5570 DATA 350,-4,78,6,351,-4,80,1,352,-4,78,6,353,-4,85,2,354,-4,79,8
5580 DATA 360,10,1,361,10,2,362,4,3,363,4,4
5590 DATA 370,-1,1,371,-1,2,372,-1,3,373,-1,4
5600 DATA 380,-2,1,381,-2,2,382,-2,3,383,-2,4
5610 DATA 390,5,0,391,5,0,392,5,0,393,5,0
5620 DATA 400,-4,96,5,401,-3,98,7,402,-3,98,8,403,-3,98,9,404,-3,98,10
5630 DATA 405,-4,98,5,406,-4,98,6,407,-4,98,7
5640 DATA 450,-3,96,5,451,-4,98,5,452,-3,98,6,453,-4,98,7
5650 DATA 455,3,1,456,2,2,457,3,3,458,2,4,459,3,5,460,2,6,461,3,7,462,2,8
5660 DATA 464,6,0,465,5,0,466,6,0,467,5,0,468,6,0,469,5,0
5670 DATA 470,10,0,472,4,0,473,10,0,474,4,0
5680 DATA 475,1,1,476,6,0,477,1,2,478,6,0,478,1,3,479,6,0
5690
5700 DATA 480,-3,123,4,484,-3,123,4,-0,-3,124,4,488,-3,123,4,-0,-3,124,4
5710 DATA 492,-3,123,4,494,-3,124,4
5720 DATA 0,10,0
5730 DATA 496,-3,123,4,00,-3,124,4,00,-3,123,8
5740 DATA 0,10,0
5750 DATA 500,-3,123,4,00,-3,123,8,00,-3,124,4,00,-3,124,8
5760 DATA 510,-3,123,4,00,-3,123,8,00,-3,124,4,00,-3,124,8
5770 DATA 520,-3,123,4,00,-3,123,8,00,-3,124,4,00,-3,124,8
5780 DATA 540,-4,134,5,540,-4,134,7
5790 DATA 545,10,0,546,10,0,547,10,0,548,10,0
5800
5810 DATA 572,-3,145,5,0,-3,146,4
5820 DATA 580,-3,145,5,581,-3,146,6,582,-3,146,4,583,-3,147,3,0,3,0
5830 DATA 584,-3,145,5,585,-3,146,6,586,-3,146,4,587,-3,147,3,0,3,0
5840 DATA 588,-3,148,2,589,-4,147,5,590,-3,147,7,591,-4,150,3,0,3,0
5850 DATA 592,-4,150,3,593,-4,148,5,594,-4,150,4,595,-4,149,5,0,3,0
5860 DATA 600,-3,149,1,601,-3,148,8,602,-3,147,3,603,-3,147,7,0,6,0
5870 DATA 610,-3,150,0,611,-3,150,10,612,-3,151,1,613,-3,151,9,0,-4,150,3
5880 DATA 614,-3,152,2,615,-3,152,8,616,-3,153,3,617,-3,153,7,0,-4,150,7
5890 DATA 618,-3,154,4,619,-3,154,6,620,-3,155,5,621,-4,152,5
5900 DATA 630,-3,148,2,631,-4,147,5,632,-4,147,7,633,-4,150,3,634,3,0
5910 DATA 635,-4,150,3,636,-4,148,5,637,-4,150,4,638,-4,149,5,639,3,0
5920 DATA 10000
5930 -----
5940 エンドオブリスト
5950 -----

```

パソコンでカセットテープの管理を

Cassette Manager

みなさんはカセットテープの整理をどうしていますか。たいていの人は箱の中にランダムに入れたままにしているのではないのでしょうか。このプログラムは、カセットテープ専用のデータベース

で、プリントアウトしたものをそのままカセットレーベルとして使用できる便利なものです。

ほとんどのプリンタで、プログラムを変更することなく使用することができます。

村田暁生

カセットテープの管理はたいへん!

好きなアーティストの曲を録音したカセットテープは、だれでも持っているでしょう。では普段、録音したカセットテープをどのようにして整理していますか。

録音するたびに、テープのレーベルをちゃんと記入しているような几帳面な人なら、聞きたい曲を即座に見つけることができるでしょうが、私のように「面倒くさいから後でいいや」なんて思っ、ちゃんと書かない人間は後から見ると「このテープには何が入っているのかなあ」と悩むはめになります。聞きたくなっても「これでもない」「あれでもない」とテープを引っかき回すことになってしまうわけです。

それに、たとえレーベルをしっかりと書いていても、テープの数が増え、100本と増えてくれば、必要なテープはなかなか見つからなくなります。

たくさんあるもの（データ）の中から必要なものを探して取り出す...といえば、これはいうまでもなくデータベースの仕事です。そんなわけで、カセットテープ管理専用のデータベース「Cassette Manager」を作りました。

持っている機能は?

「Cassette Manager」は、アルバム名、曲名・アーティスト名（A・B面10曲ずつ）を、最大200本まで管理できます。

アルバム名は36文字×2行まで、曲名は

23文字まで、アーティスト名は10文字までです。実際に使うと、ちょっと足りないのですが、自分で分かりやすいように省略したり、記号を決めたりして使ってみてください。

データベースというわけですから、もちろん検索機能がついています。アルバム名、曲名、アーティスト名、登録日時など、どの項目でも検索させることができます。たとえば、「"なんとか by the sea" はどこに入ってたっけ」とか、「たしか」HOW なんとか KNOW" だったかな?」といったように、名前がウロ覚えでも検索できるようになっています。

また、入力したデータを元に、レーベル

をプリンタに打ち出すこともできます。プリンタの種類は問いません。ただし、アルバム名は35文字×2行、曲名は19文字だけ、アーティスト名は打ち出せません。いってみれば、レーベル作りも全部パソコンに任せてしまいたい、という横着な人のための機能です。もっと面倒くさがりな人は、ケースにシリアルナンバーを大きく書いておくだけでも十分でしょう。

準備をしなけりゃ始まらない

まず、後ろにあるリスト1を打ち込んでください。とにかく、これをしなければ何もできません。全部で300行以上ありますから

図1 プリントアウトしたレーベル

Side A	Side B
HOW WILL I KNOW ALL AT ONCE TAKE GOOD CARE OF ME GREATEST LOVE OF ALL HOLD ME	YOU GIVE GOOD LOVE THINKING ABOUT YOU SOMEONE FOR ME SAVING ALL MY LOVE NOBODY LOVES ME LIKE

```

┌ NO. YOU GIVE GOOD LOVE
└ 003 WHITNEY HOUSTON

┌ DATE 85/05/24 TIME 02:24:33
└ SOURCE ----- BIAS EQ-----
  NR IN ☐ OUT ☐
  STEREO ☐ MONO ☐
    
```


ら、打ち間違には注意しましょう。

プログラムは、全部 BASIC です。特殊な命令は使っていないので、PC-8800 シリーズの全機種で使えると思います。ただし、フロッピーディスクドライブは必需品ですので注意してください。

インデックスを打ち出すには、プリンタが必要です。PC シリーズ用のプリンタなら、プリンタ制御コマンドを送らない限り文字の間隔が等しいはずですから、すべて使用可能（のはず）です。

プログラムに打ち込んだら、フロッピーディスクにセーブしましょう。そして、データディスクとして、フォーマットしたフロッピーディスクを1枚用意してください。「Cassette Manager」はデータが最大で150K バイトになりますので、これだけ空き領域があれば新しいフロッピーディスクを使わなくても構いません。

最後に、内蔵クロックの時間を合わせておってください。登録日時、中の時計を参考しますので、合っていないとんでもない日づけや時刻になってしまいます。

（ プログラムを 走らせると… ）

リストをすべて打ち込んで、データディスクが用意できた実行してみましょう。

まず、画面を作成しますのでしばらく待っていてください（FR で約5秒）。ぱっと画面が表示され、「データディスクをドライブ1に入れてください」というメッセージが出ます。データディスクをセットしたら、キーボードを叩いてください。どのキーでも構いません。これから終了するまでは、絶対にデータディスクを取り替えない取り出したりしないようにしてください。データが破壊されてしまいます。

画面は、左上の大きな空欄がアルバム名、中央が曲名、アーティスト名、右上がシリアルナンバーと登録日時です。だいたい、ひとめ見ればすぐに理解できるでしょう。

画面の下には、各種コマンドが表示されています。ここで使えるのは、0～4 の数字と上下のカーソルキーだけです。なお、このように各種コマンドが表示されている画面を「基本画面」と呼ぶことにします。すべてのコマンドは基本画面から選択することになります。

それでは、各コマンドを説明しましょう。

（ コマンドを理解しよう ）

① ツイカ

データを追加するコマンドです。新しいデータを、現在あるデータのいちばん後ろに追加します。

基本画面で「1」を押してください。画面にデータが表示されている場合は消され、カーソルがアルバム名の欄で点滅します。キーボードからデータを入力してください。DEL キーを押すと、カーソルの前の1文字が消去されます。カーソルキーでそれぞれの方向に入力欄が移ります。RETURN キーは、アルバム名→A 面1曲目の曲名→A 面1曲目のアーティスト名→A 面2曲目の曲名→…、という具合に入力欄が移ります。これと逆の順序で入力欄を移動したい場合は、CTRL キーを押しながら H を押してください。

データをすべて入力したら TAB キーを押してフロッピーディスクに登録します。中断したい場合は、ESC キーを押して基本画面にもどり、最初からやり直してください。

② ヘンコウ

データの入力を間違ってしまった場合や、録音したテープに別の曲をかぶせた場合に使うコマンドです。基本画面で上下のカーソルキーを使い、目的のテープのデータが表示されたら「2」を押してください。

入力方法は、「1 ツイカ」とまったく同じです。カーソルキーなどを使って、変更してください。変更後の結果をディスクに登録するときは、TAB キーを押します。中断するときは ESC キーを押してください。

③ インサツ

入力したデータをプリンタに印刷するコマンドです。「2 ヘンコウ」と同様、基本画面でカーソルキーを使い、打ち出したいデータを表示させてから「3」を押してください。

プリンタが正しく接続できたら、TAB キーを押します。プリンタが接続できないときなどは ESC キーを押してください。

すでに述べたとおり、基本的にはプリンタを選ばないプログラムになっています。そのために、少々大きさがいい加減になっています（PC-PR201の場合は少し縦長）。印刷されるガイドは、あくまで目安として使ってください。実際のテープインデックスを規定に使って cutter で切り取れば簡単にいいと思います。

また、アルバム名、曲名ともに後ろのほうで切れてしまいます。

不満な方は、リストの3590行から3770行をお手持ちのプリンタに合わせて変更してください。

④ ケンサク

キーワードを入力し、そのキーワードが含まれるデータを検索します。入力したキーワードは、アルバム名、曲名、アーティスト名、登録年月日のすべての欄を対象として検索してくれるわけです。

基本画面で「4」を押すと、キーワードを開いてきます。キーボードから入力してください。RETURN キーを押すと検索を始め、終われば基本画面にもどります。検索結果は、直接表示されるわけではありません。カーソルキーで前後のデータを表示させた場合、この検索から漏れたデータは表示されなくなるのです。

重ねて検索を選んだときには、「前回の検索結果を有効にしますか」と尋ねます。前回の検索結果の中から、さらに検索する場合は「Y」、無効にしてすべての中から検索する場合は「N」と入力してください。あとは、最初の検索と同じです。

たとえば「なんとか by the sea」を探したいときは「by the sea」をキーワードとして検索します。また「HOW なんとか KNOW」を探したいときは、まず「HOW」をキーワードにして検索し、次に前回の検索結果を有効にして「KNOW」で検索するわけです。

⑤ シュウリョウ

プログラムを終了します。データディスクを取り出す場合は、必ずこのコマンドを実行してください。

終了してもいいときは「Y」、基本画面にもどるときは「N」を押します。

```

1000 ' =====
1010 '                      Cassette Manager
1020 '                      data base for compact cassette
1030 ' =====
1040 SCREEN 2,3
1050 CONSOLE 0,25,0,1
1060 WIDTH 80,20
1070 CLS 2
1080 DEFINT A-Z
1090 FIELD #1,255 AS RC$,1 AS DY$
1100 DIM DAT$(43),N1(199),N2(199),P(41,2)
1110 LINE(4,37)-(316,77),,B
1120 LINE(4,117)-(628,317),,B
1130 LINE(316,117)-(316,317)
1140 FOR I=137 TO 297 STEP 20
1150 LINE(4,I)-(628,I),,SHA000
1160 NEXT
1170 LINE(212,117)-(212,317),,SHA000
1180 LINE(528,117)-(528,317),,SHA000
1190 PAINT(0,0),CHR$(65535)+CHR$(0),7
1200 LINE(4,20)-(43,36),0,BF
1210 LINE(576,18)-(626,34),0,BF
1220 LINE(558,38)-(626,54),0,BF
1230 LINE(558,58)-(626,74),0,BF
1240 LINE(0,337)-(639,399),0,BF
1250 LINE(4,100)-(50,115),0,BF
1260 LINE(320,100)-(366,115),0,BF
1270 FOR I=1 TO 5
1280 READ A$,
1290 PUT(A,21),KANJI(ASC(A$)),OR
1300 PUT(A+1,21),KANJI(ASC(A$)),OR
1310 NEXT
1320 DATA 5,T,13,1,20,t,28,1,35,e
1330 FOR I=1 TO 3
1340 READ A$,
1350 PUT(A,28),KANJI(ASC(A$)),OR
1360 PUT(A+1,28),KANJI(ASC(A$)),OR
1370 NEXT
1380 DATA 578,N,586,o,594,,
1390 FOR I=1 TO 10
1400 READ A$,
1410 PUT(A,101),KANJI(ASC(A$)),OR
1420 PUT(A+1,101),KANJI(ASC(A$)),OR
1430 NEXT
1440 DATA 5,S,13,1,20,d,29,e,41,A,321,S,329,1,336,d,345,e,357,B
1450 SCREEN 0
1460 LOCATE 75,1:PRINT "000"
1470 LOCATE 78,2:PRINT "00:00/00"
1480 LOCATE 78,3:PRINT "00:00:00"
1490 CONSOLE 17,3
1500 CLS
1510 PRINT "DATA DISK ヲ DRIVE 1 ニ イレ ク"キキ。":
1520 A$=INPUT$(1)
1530 PRINT
1540 OPEN "casset.dat" AS #1
1550 MAX=LOF(1)/3
1560 NDX=MAX
1570 FOR I=0 TO 199
1580 NI(I)=I
1590 NEXT
1600 FOR I=0 TO 41
1610 FOR J=0 TO 2
1620 READ P(I,J)
1630 NEXT
1640 DAT$(I)=STRING$(P(I,2),SHA000)
1650 NEXT
1660 DAT$(42)="00:00/00"
1670 DAT$(43)="00:00:00"
1680 DATA 2,2,36,2,3,36,2,6,23,2,7,23,2,8,23,2,9,23,2,10,23,2,11,23,2,12,23
1690 DATA 2,13,23,2,14,23,2,15,23,28,6,10,28,7,10,28,8,10,28,9,10,28,10,10
1700 DATA 28,11,10,28,12,10,28,13,10,28,14,10,28,15,10,41,6,23,41,7,23,41,8,23
1710 DATA 41,9,23,41,10,23,41,11,23,41,12,23,41,13,23,41,14,23,41,15,23,67,6,10
1720 DATA 67,7,10,67,8,10,67,9,10,67,10,10,67,11,10,67,12,10,67,13,10,67,14,10
1730 DATA 67,15,10
1740 GOTO 2898
1750 GGET
1760 GET #1,REC#3-2
1770 MID$(DAT$(42),1,8)=MID$(RC$,1,8)
1780 MID$(DAT$(43),1,8)=MID$(RC$,9,8)
1790 MID$(DAT$(0),1,36)=MID$(RC$,17,36)
1800 MID$(DAT$(1),1,36)=MID$(RC$,53,36)
1810 FOR I=0 TO 9
1820 MID$(DAT$(I+12),1,10)=MID$(RC$,89+I*10,10)
1830 NEXT
1840 FOR I=0 TO 5
1850 MID$(DAT$(I+32),1,10)=MID$(RC$,189+I*10,10)
1860 NEXT
1870 GET #1,REC#3-1
1880 FOR I=0 TO 3
1890 MID$(DAT$(I+38),1,10)=MID$(RC$,1+I*10,10)
1900 NEXT
1910 FOR I=0 TO 8
1920 MID$(DAT$(I+21),1,23)=MID$(RC$,41+I*23,23)
1930 NEXT
1940 GET #1,REC#3
1950 MID$(DAT$(11),1,23)=MID$(RC$,1,23)
1960 FOR I=1 TO 10
1970 MID$(DAT$(I+21),1,23)=MID$(RC$,1+I*23,23)
1980 NEXT
1990 RETURN
2000 =PUT
2010 MID$(RC$,1,255)=STRING$(255,SHA000)
2020 MID$(RC$,1,8)=DAT$(42)
2030 MID$(RC$,9,8)=DAT$(43)
2040 MID$(RC$,17,36)=DAT$(0)
2050 MID$(RC$,53,36)=DAT$(1)
2060 FOR I=0 TO 9

```

```

2070 MID$(RC$,89+I*10,10)=DAT$(I+12)
2080 NEXT
2090 FOR I=0 TO 5
2100 MID$(RC$,189+I*10,10)=DAT$(I+32)
2110 NEXT
2120 PUT #1,REC=3-2
2130 MID$(RC$,1,255)=STRING$(255,&HA0)
2140 FOR I=0 TO 3
2150 MID$(RC$,I+1*10,10)=DAT$(I+38)
2160 NEXT
2170 FOR I=0 TO 8
2180 MID$(RC$,4I+I*23,23)=DAT$(I+2)
2190 NEXT
2200 PUT #1,REC=3-1
2210 MID$(RC$,1,255)=STRING$(255,&HA0)
2220 MID$(RC$,1,23)=DAT$(1)
2230 FOR I=1 TO 10
2240 MID$(RC$,I+1*23,23)=DAT$(I+21)
2250 NEXT
2260 PUT #1,REC=3
2270 RETURN
2280 *INP
2290 C=0:Y=0
2300 LOCATE P(C,0),P(C,1):PRINT DAT$(C):
2310 X=INSTR(DAT$(C),CHR$(&HA0))-1
2320 IF X<0 THEN X=LEN(DAT$(C))
2330 LOCATE P(C,0)+X,P(C,1)
2340 I=ASC(INPUT$(1))
2350 IF I= 8 THEN 2030
2360 IF I= 9 THEN RETURN
2370 IF I= 11 THEN 2010
2380 IF I= 12 THEN 2010
2390 IF I= 13 THEN 2550
2400 IF I= 27 THEN RETURN
2410 IF I= 28 THEN 2010
2420 IF I= 29 THEN 2050
2430 IF I= 30 THEN 2090
2440 IF I= 31 THEN 2750
2450 IF I=127 THEN 2520
2460 IF I< 32 OR X=LEN(DAT$(C)) THEN 2300
2470 MID$(DAT$(C),X+1,1)=CHR$(I)
2480 IF X=LEN(DAT$(C))-1 THEN 2500
2490 GOTO 2300
2500 LOCATE P(C,0),P(C,1):PRINT DAT$(C):
2510 GOTO 2550
2520 IF X=0 THEN 2300
2530 MID$(DAT$(C),X,1)=CHR$(&HA0)
2540 GOTO 2300
2550 IF C<2 THEN C=C+1:GOTO 2300
2560 IF C=41 THEN C=0:GOTO 2300
2570 IF C=21 THEN C=22:GOTO 2300
2580 IF (C-2)*10=0 OR (C-2)*10=2 THEN C=C+10:GOTO 2300
2590 C=C-9
2600 GOTO 2300
2610 IF C<2 THEN 2300
2620 IF C<31 THEN C=C-30:GOTO 2300
2630 C=C+10
2640 GOTO 2300
2650 IF C<12 THEN C=C+30:GOTO 2300
2660 C=C-10
2670 GOTO 2300
2680 IF C=0 THEN C=1:GOTO 2300
2690 IF C=12 THEN C=1:GOTO 2300
2700 IF C=22 THEN C=31:GOTO 2300
2710 IF C=22 THEN C=31:GOTO 2300
2720 IF C=32 THEN C=41:GOTO 2300
2730 C=C-1
2740 GOTO 2300
2750 IF C=11 THEN C=0:GOTO 2300
2760 IF C=21 THEN C=0:GOTO 2300
2770 IF C=31 THEN C=22:GOTO 2300
2780 IF C=41 THEN C=32:GOTO 2300
2790 C=C+1
2800 GOTO 2300
2810 C=0
2820 GOTO 2300
2830 IF C=0 THEN C=41:GOTO 2300
2840 IF C=22 THEN C=21:GOTO 2300
2850 IF C<3 THEN C=C-1:GOTO 2300
2860 IF (C-2)*10=0 OR (C-2)*10=2 THEN C=C+9:GOTO 2300
2870 C=C-10
2880 GOTO 2300
2890 IF MAX=0 THEN N=0:GOTO 3040
2900 CLS
2910 PRINT "1)ヤカ 2)ハンカ 3)カネツク 4)カンツク 0)シヤツ/ヨウ )アヒ 19500 : ";
2920 POKE &HFB9,&H1E
2930 POKE &HFB0,&H1F
2940 AS=INPUT$(1)
2950 ON INSTR(CHR$(&H1E)+CHR$(&H1F)+"12340",AS) GOTO 2970,3010,3140,3370,3490,37
90,4000
2960 GOTO 2940
2970 IF MAX=0 THEN 2940
2980 N=N-1
2990 IF N<0 THEN N=0:GOTO 2940
3000 GOTO 3040
3010 IF MAX=0 THEN 2940
3020 N=N+1
3030 IF N>=NMX THEN N=NMX-1:GOTO 2940
3040 REC=N(I):I=GOSUB *GET
3050 LOCATE 75,1:PRINT RIGHT$( "00"+MID$(STR$(N(I)+1),2),3)
3060 LOCATE 70,2:PRINT DAT$(42)
3070 LOCATE 70,3:PRINT DAT$(43)
3080 LOCATE P(0,0),P(0,1):PRINT DAT$(0);
3090 LOCATE P(1,0),P(1,1):PRINT DAT$(1);
3100 FOR I=2 TO 11
3110 LOCATE P(I,0),P(I,1):PRINT DAT$(I); " ";DAT$(I+10); " ";DAT$(I+20); "
";DAT$(I+30);

```



```

3120 NEXT
3130 GOTO 2900
3140 CLS
3150 PRINT "DATA ライカ シマス。"
3160 PRINT "カメラ- デ カ-カメラ、TAB+ デ カ+コピ、ESC+ デ チョクタン デス。"
3170 MID$(DAT$(42),1,8)=DATE$
3180 MID$(DAT$(43),1,8)=TIME$
3190 LOCATE 75,1:PRINT RIGHT$("00"+MID$(STR$(MAX+1),2),3)
3200 LOCATE 70,2:PRINT DAT$(42)
3210 LOCATE 70,3:PRINT DAT$(43)
3220 FOR I=0 TO 41
3230 MID$(DAT$(1),1,23)=STRING$(P(1,2),&H0)
3240 LOCATE P(1,0),P(1,1):PRINT DAT$(1);
3250 NEXT
3260 GOSUB #1NP.
3270 IF I=27 THEN 3320
3280 REC=MAX+1
3290 GOSUB #PUT.
3300 N=MAX
3310 MAX=MAX+1
3320 FOR I=0 TO 199
3330 NI(1)=I
3340 NEXT
3350 NM=MAX
3360 GOTO 3040
3370 IF MAX=0 THEN 2940
3380 CLS
3390 PRINT "DATA ランコウ シマス。"
3400 PRINT "カメラ- デ カ-カメラ、TAB+ デ カ+コピ、ESC+ デ チョクタン デス。"
3410 MID$(DAT$(42),1,8)=DATE$
3420 MID$(DAT$(43),1,8)=TIME$
3430 LOCATE 70,2:PRINT DAT$(42)
3440 LOCATE 70,3:PRINT DAT$(43)
3450 GOSUB #1NP.
3460 IF I=27 THEN 3320
3470 GOSUB #PUT.
3480 GOTO 3320
3490 #PRT.
3500 IF MAX=0 THEN 2940
3510 CLS
3520 PRINT "INDEX を PRINT OUT シマス。"
3530 PRINT "ファンク / シュビ" カ デキタラ TAB+ ラ キタタキタイ、ESC+ デ チョクタン デス。";
3540 AS=INPUT(1)
3550 IF AS=CHR$(27) THEN 2900
3560 IF AS<CHR$(9) THEN 3540
3570 PRINT
3580 PRINT "インサツ チョウ デス。"
3590 LPRINT
3600 LPRINT "r":SPC(39);"v"
3610 LPRINT
3620 LPRINT " Side A Side B"
3630 LPRINT "-----"
3640 FOR I=0 TO 9
3650 LPRINT " ":LEFT$(DAT$(1+2),19); " ":LEFT$(DAT$(1+22),19)
3660 NEXT
3670 LPRINT
3680 LPRINT
3690 LPRINT "t":SPC(39);"t"
3700 LPRINT "No. ":LEFT$(DAT$(0),35)
3710 LPRINT " ":RIGHT$("00"+MID$(STR$(REC),2),3); " ":LEFT$(DAT$(1),35)
3720 LPRINT "t":SPC(39);"t"
3730 LPRINT "DATE ":DAT$(42);" TIME ":DAT$(43)
3740 LPRINT "SOURCE BIAS EQ-----"
3750 LPRINT "NR IN 0 OUT 0"
3760 LPRINT "STEREO 0 MONO 0"
3770 LPRINT "s":SPC(39);"j"
3780 GOTO 2900
3790 IF MAX=0 THEN 2940
3800 CLS
3810 PRINT "DATA ランコウ シマス。"
3820 IF MAX=NM THEN 3930
3830 PRINT "ランコウ / ランコウ デ カ-カメラ (Y/N) : ";
3840 ON INSTR("yYnNzZ",INPUT$(1))%3 GOTO 3920,3860
3850 GOTO 3840
3860 PRINT "No"
3870 FOR I=0 TO 199
3880 NI(1)=I
3890 NEXT
3900 NM=MAX
3910 GOTO 3930
3920 PRINT "Yes"
3930 LINE INPUT "ランコウ & KEYWORD n? : ";K$
3940 NN=0
3950 FOR I=0 TO NM-1
3960 REC=NI(1)+1
3970 FOR J=0 TO 2
3980 GET #1,REC=J
3990 IF INSTR(RC$,K$) THEN N2(NN)=I:NN=NN+1:J=J+1
4000 NEXT
4010 NEXT
4020 FOR I=0 TO NN-1
4030 NI(1)=N2(1)
4040 NEXT
4050 NM=NN
4060 N=0
4070 GOTO 3040
4080 CLOSE
4090 CLS
4100 PRINT "ランコウ シテ イデ-ズ? (Y/N) : ";
4110 ON INSTR("yYnNzZ",INPUT$(1))%3 GOTO 4130,2900
4120 GOTO 4110
4130 PRINT "Yes"
4140 CONSOLE 0,25
4150 END

```

ギャンブルをパソコンで楽しむ ボートレース

米田 稔一

ギャンブルは麻雀のようなもの、それで身を滅ぼす人のなんと多いこと。でも昨今の投機ブームばかり、ギャンブル性というのは人間の性かもしれません。でもせつかく身を削るような思いをして貯めたお金を一瞬のうちになくしてしまうというのも情ないものです。お金を投じずにギャンブルでできれば、それに越したことはありません。このプログラムはそんな方のために作られた、本物さながらのボートレースゲームです。

ギャンブルの薦め!?

今のパソコンゲームの主流というところ、シューティングゲームにロールプレイングゲームですが、どうも殺伐としたものが多過ぎるようです。雲霞のごとく現れる敵のキャラクターをやみくもに叩き落とすだけのシューティングゲームとか、迷宮や荒野をあてどもなくさまよい、追い剥ぎ同然の所業(それまで平和に暮らしていたモンスターに何の恨みがあるのでしょうか)を重ねて経験値を稼ぐといったロールプレイングゲームに、貴重な時間を浪費するのはもったいない話で、時間はもっと有効に使いたいのです。

「時は金なり。世の中、ゼニや!」と大阪商人のド根性にも似て、最近大人たちの間で人気がマナーゲーム。低金利時代の今、銀行預金だけだと利殖には不利と、株とか先物取引でひと儲けをたくらむ方も多そうです。しかしこれも所詮はギャンブルのようなもの。NTT株で大儲けしたと思ったら、株式市場の大暴落で大損夜逃げ。素人衆にはチト難しいようです。そこで奥の手に登場するのが公営ギャンブルです。

現在、公営ギャンブルとして認められているのが、競輪、競馬、ボートレースの3種類。いずれも複数の馬や自転車、ボートが出走して、着順を予想します。予想が当たれば、配当金がもらえます。

ほかにもパチンコ、麻雀、花札、サイコ

ロ、野球賭博といった私営ギャンブルが存在するかもしれませんが、このテの私営ギャンブルは参加するだけで警察の厄介になりますので、くれぐれも手を出さないようにしてください。

とはいってもあなたが未成年者ならば公営ギャンブルに手を出すのも法律違反。また、あなたが社会人ていくらか自分が稼いだお金とはいえ、買った馬券がはずれて一瞬のうちに紙クズになってしまうのは空しいもの。「ああ、こんなことならばビデオを買っておくべきだったなあ…」と悔やんだことも…私の友だちの話です。

いかに世の中は諸行無常といっても、ビデオが紙クズに化けてはたまりません。そこで今回は、いくらか予想が外れても懐に響かないように、もとい、レース場に行けない未成年者のために、ボートレースをパソコンゲームにしてみました。これなら(懐に關係なく)安心して、ボートレースの楽しさを味わうことができます。それ、2-3で決まりやあ! あらら…。

必要なハードウェア

- ① PC-8800シリーズのパソコン本体
- ② フロッピードライブ (最低1台必要)
- ③ カラーディスプレイ (低解像度でも可)

プログラムの 入力と実行方法

- ① フォーマットしたフロッピーを1枚用

意します

- ② DISK-BASICを起動します。このゲームはV1モードで動作します
- ③ リストAを入力してセーブします
SAVE "BOAT"◎

と入力してください

- ④ リストBはボートのキャラクターデータです

MON◎

でマシン語モニタを起動して、
h]

が表示されたら、

EBC00◎(BC00番地から編集開始)
を入力します

- ⑤ リストBを入力したら、
STOPキーを押して、再び"h]"が表示されたら、

CTRL+Bキーを押してBASICにもどります。そして、

BSAVE "B", &HBC00, &H2400◎
でリストBをファイル名"B"でセーブします

- ⑥ リストCは背景、文字、数字のデータです

リストBと同様にマシン語モニタを起動して、
EBC00◎

でBC00番地から入力を始めます

- ⑦ リストCの入力が終わったら、
BSAVE "C", &HBC00, &H1760◎
でリストCをファイル名"C"でセーブします

- ⑧リストDはマシン語のメインプログラムの前半です。リストBと同様にマシン語モニタを起動して、

EBD20 ⊕

でBD20番地から入力を始めます

- ⑨リストDの入力が終わったら、

BSAVE "D", &HBD20, &HD20 ⊕

でリストDをファイル名"D"でセーブします

- ⑩リストEはマシン語のメインプログラムの後半です。リストBと同様にマシン語モニタを起動して、

ECB00 ⊕

でCB00番地から入力を始めます

- ⑪リストEの入力が終わったら、

BSAVE "E", &HCB00, &HB00 ⊕

でリストEをファイル名"E"でセーブします。これで入力は終了です

- ⑫リストを間違いない入力しているのを確認したら、

RUN "BOAT" ⊕

でゲーム開始です

- ②画面中央に各コースのボートの番号と勝率が表示されます。勝率が「高いほど1, 2着になる可能性が高いわけです」

- ③勝率と持ち金をもとに予想します。1, 2着になりそうだな、という番号を連番に入力します

1～6の数字が入力可能です。たとえば、1着が3番、2着が5番だと予想したら、「3-5」と指定します

ただし、「3-3」のように同じ番号を指定することはできません

- ④つぎにその組み合わせにくらつぎ込む、もとい、何円投資するか入力します。投資できる金額は100円から999,900円までです

- ⑤1～5レースは連勝複式、6～10レースは連勝単式です。

連勝複式とは、上位2着の組み合わせを当てるもので、1着と2着が入れ替わっていてもかまいません。たとえば1着6番、2着4番の場合、「6-4」でも、「4-6」でも当たりです

連勝単式とは、連勝複式と違って1着と2着の番号が入れ替わっている場合はハズレです。上の例だと「6-4」は当たりですが、「4-6」はハズレです

- ⑥各レースごとに5種類の予想ができます。5種類予想するとレースが始まりますが、1種類から4種類でレースをスタートしたい場合は、スペースキーを押してください

- ⑦つぎに配当金額が表示されます。配当金額というのは、予想が当たった場合、100円につきいくら返ってくるかという金額のことです。つまり、配当金額が1,200円で200円投資していた場合、

$$(200 \div 100) \times 1,200 = 2,400$$

で、2,400円返ってきます。ただし正確には馬券を買ったお金は差し引かれますので、純利益は、

$$2,400 - 200 = 2,200$$

で、2,200円です
配当が10,000円を超える場合、万馬券(ボートレースでも万「馬」券です)とい

います。なにしろ、100円が10,000円になりますので、ウハウハ喜ぶということになりますが、あくまでも予想が当たったときの話。取らぬ狸の皮算用にならないように注意しましょう

- ⑧スペースキーを押すとレース開始です。画面には、レース番号と、距離、時計、そしてコースとボートが表示されます。コース中央のラインがスタート/ゴールラインです

時計がちょうど12時を差したらスタートです。コースは1周400メートルで、反時計回りに展開します

- ⑨自分の予想が当たるよう、必死に念じましょう。

また、ときどきフライングといって、レースから脱落するボートがあります。この番号は画面下の、「フライング」のところに表示されます

- ⑩1,2着の艇がゴールラインを過ぎるとレース終了です。1,2着のコース番号と配当が表示されます。予想が当たったときはムラムラと喜びがこみ上げてきます(というのは私だけでしょうか)。

また、ときどき画面では1,2着がゴールラインを越えているのに、3着以降がゴールを過ぎないとレースが終了しないときがあります。これは1,2着のいずれかの艇が反則行為を行ったためで、この艇は失格となります。正式な着順は、配当といっしょに表示されます。

- ⑪スペースキーを押すと、配当と持ち金が表示されます

「配当」がいまのレースで当たった金額、「掛金」が予想に投資した金額です。また、現在の持ち金と配当、掛金を清算したものが、「計」として表示されます。

もし持ち金がマイナスの場合は、「持ち金」ではなく、「借金」が赤字で表示されます

- ⑫計がいの持ち金です。持ち金が画面右下に表示されます。

注意しないといけないことは、「悪銭身につかず」の教えどおり、10,000,000円以上の単位の金額は、強制的にカットされることです。しかし10,000,000円以上の借金も棒引きされるので、「人間万事塞翁が馬」ということかもしれません。こうして、また各レースごとに①から繰り返します。

ボートレース的中必勝法

ボートレースなんてものは、ただやみく

ボートレースのルール

このボートレースは6そう立てで出艇します。各艇には1～6の番号がついています。

コースは1周400メートルで、このコースを3周から5周して順位を決定します。

レースは全10レースで1～5レースは「連勝複式」(上位2着の番号を当てるもので、1着と2着の順番が入れ替わっていてもOK)の1,200メートルレース(コースを3周)、6～10レースは「連勝単式」(1着、2着の順番が正しく含まなくてはだめ)の2,000メートルレース(コースを5周)です。

あなたは観客となって順位を予想します。最初の持ち金は500,000円(〆)。各レースごとに5種類の予想を立てることができます。

万一、持ち金がマイナスになった場合は「借金」をしてゲームを続けることができます。

操作方法

- ①各レースが始まると、まず予想を行います

もに資金をつぎ込んでも、なかなか当たる
ものではありません。やはり、冷静に分析
して予想したいもの。そうでなければ「借
金地獄」で喘ぐことにもなりかねません。
そこで、ボートレースの必勝法を考えてみ
ました。

CHECK 1

勝率をチェックせよ！

予想の基礎となるのは、なんといっても「勝率」です。30%の勝率のボートと、50%の勝率のボートだと、どちらが勝ちやすいか一目瞭然です。

まず、勝率の高い順番にボートを並べてみます。そして、勝率が1番目に高いコース（以後、本命と呼びます）と2番目に高いコース（以後、対抗と呼びます）を選んでみましょう。もしこの2つの勝率がほかのコースにくらべて圧倒的に高い場合（たとえば、上2つが50%、その他は30%台のときなど）は、この2艇が1、2着に入る可能性は非常に高くなります。とくに連勝複式のレースの場合、どちらが1着でも当たりですので当てやすいレースだといえます。

CHECK 2

本命を軸に流す

もし本命だけ強く、ほかがどんぐりの背くらべというときは、本命を軸に複数枚に振り分ける方法があります。

たとえば、2コースが圧倒的に強く、ほかのコースは同じような勝率の場合には、「2-1」、「2-3」、「2-4」、「2-5」、「2-6」と選んでみましょう。投資する額は多くなり、かならず4つはハズレますが、高配当がつけば、そんなものは何てもありません。

CHECK 3

連勝単式はクロスで狙う

連勝単式は、連勝複式と違って順位も当てなければならぬので、当たりにくいレースです。でもそれだけに当たったときの配当も大きいもの。やはり連勝単式も、本命と対抗の組み合わせで選ぶのがベターですが、そのときは必ずクロスで狙いましょう。

クロスというのは、「本命-対抗」、「対抗-本命」の組み合わせで選ぶことです。たとえば、本命が1コース、対抗が5コースの場合、「1-5」、「5-1」で選びます。これ

なら、本命と対抗の順位が逆でも、当てることができる。

CHECK 4

休むのも手の内

とにかく毎レース毎レース賭けなくては
気が済まない方がいますが、別に全レース
に賭ける必要はどこにもありません。とく
にどのコースの勝率も変わりばえないと
きは、思い切って賭けるのをやめましょう。
そして、狙ったレースがくるまで、じっと
我慢するのです。ここぞというレースで全
力を注ぐのが、損をしない方法です。

CHECK 5

ギャンブルに「絶対」はない

いまだに勝率を重視する方法を紹介しましたが、勝率はあくまで指標ということをお忘れのように。本命、対抗が強く、ガチの銀行レースだと思っていたら、フラグと反則行為で上位2艇が失格し、とんでもない組み合わせが入ったということもよくあることです。ギャンブルに絶対は

ありません。

番外編
マル秘情報

「本命を軸に流して狙え」と書きましたが、連番の左側に同じ番号を2つ続けるとその艇は遅くなり、順位が下がる人が多いようです。その場合は、「2-1」、「3-2」、「2-4」、「5-2」、「2-6」と散らして選びましょう。

ただし連勝複式のレースの場合、ですよ。

リストA

```

100 ' ホート レース
110 BLOAD "B",&H1000
120 BLOAD "C",&H3C00
130 BLOAD "D"
140 BLOAD "E"
150 DEF USR1=&HC000
160 I=USR1(I)
170 END

```

リストB

[illegible]

ADDR	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#A	#B	#C	#D	#E	#F	STATUS
ADDR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	172
DATA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	173
DATA	87	FC	87	E2	8C	1F	08	1C	00	7F	82	3F	80	80	80	80	174
DATA	87	FC	87	E2	8C	1F	08	1C	00	7F	82	3F	80	80	80	80	175
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	176
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	177
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	178
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	179
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	180
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	181
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	182
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	183
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	184
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	185
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	186
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	187
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	188
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	189
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	190
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	191
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	192
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	193
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	194
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	195
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	196
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	197
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	198
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	199
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	200
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	201
DATA	82	80	FF	FA	88	3F	FF	1C	00	7F	82	3F	80	80	80	80	202
DATA	82																

[illegible]

[illegible]

VSUM D3 01 75 A1 BB AB 77 FB 10 BF 19 D3 44 87 F8 70 :B0

[illegible]

VSUM 5A 0A CE 22 AF 6B FA 60 A0 64 28 7B 46 3A 0B 24 :13

[illegible]

```
VSIM C7 D5 BB 27 08 00 EB 60 76 17 33 90 7C 03 95 1C :0
```

ADD#	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+5U
C580	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C510	00	00	00	00	0E	1C	38	00	1E	1C	38	00	0E	1C	38	00	00
C520	0E	1C	38	00	03	F7	F0	00	00	00	00	00	00	00	00	00	00
C530	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C540	18	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00
C550	FF	FF	FF	FF	01	E3	C7	E8	E1	E3	C7	FC	01	E3	C7	FF	FF
C560	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C570	00	00	00	00	00	20	0C	00	00	00	00	00	00	00	00	00	00
C580	00	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00
C590	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C5A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C5B0	00	00	00	00	00	20	04	00	00	00	00	00	00	00	00	00	00
C5C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C5D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C5E0	02	00	00	00	00	03	58	00	10	01	F8	00	00	30	7C	00	00
C5F0	00	1F	1E	82	01	C7	FC	00	01	F1	FB	00	00	7F	68	00	00

```
*****  
USFHA 08 06 16 05 09 0C 74 0A 2D F9 EC 0C 71 04 2B ED :0
```

[illegible]

```

C9F0 01 FF FE 00 01 FF FE 42 11 FF FE 00 01 0C 00 00 00
-----

```

[illegible]

CAFE 04 00 0E 00 00 00 38 02 00 00 00 00 00 01 00 .

[illegible][illegible]

```
VSUM 2F 03 19 9C 4F E7 FA 7A 60 88 F9 F5 26 40 EE 40 :7B
```

[illegible]

```
VSUM 58 32 4B 39 63 9B AE 2D C5 80 F9 66 37 C8 E6 68 :D8
```

リストC

[illegible]

```
VSUM 04 00 40 08 04 00 40 10 04 80 48 12 00 00 00 00 :7E
```

[illegible]

```
VSUM 05 0D 0D 04 EC 9C 24 05 7E A6 28 F7 F7 F7 F7 :AA
```

リストC

```

-----
YSUM FB FC FA FA FA F7 7A FA FB FA 7B FB FB FB 7A FA :23
-----

```

```
VSUM 7C 09 E2 28 D9 40 F1 00 E0 FF F8 A0 BC 1B D8 30 :EF
```

```
VSUM 93 16 F6 B8 97 CB DC 18 BD B8 A8 F0 9A BB BB 2A :F0
```

```
VSUM 7C 06 5C 20 DC 00 40 80 EC 04 00 00 EC 00 AC 48 :EA
```

```
VSUM 04 01 03 9C A5 0E D9 58 C9 A2 E6 30 EE 22 86 FC :9B
```

```
VSUM 7C A5 DC 28 DC 8B 8B 8B EF E6 EE 00 C8 4E EC 48 :21
```

```
VSUM BC 1B D8 30 E0 FF F8 A0 D9 40 F1 00 7C 09 E2 28 :EF
```

```
VSIM AC 13 8B CB EA 1A 20 EA EA 26 DE 20 AB 19 8B BA :1E
```

```
VSUM BC 1B EC 78 EC FF C3 C0 DC 3F 3F 80 7C 0B FB 2B :2A
```

```
-----
VSIM BB D1 E4 48 EB CE EE E9 D9 16 EC 8B 7C ED EC 28 :D8
```

```
VSUM 9F 8A 8A 38 E4 3F 3F C8 E7 FE 83 88 9E 1A EB 68 1A?
```

```
VSUM 40 13 0C 26 F0 C9 09 42 EE 19 86 42 40 ED 42 16 10E
```

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CA00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 83 83 83 83 83 83 86 86 86 86 83 83 83 83 83 152

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CB00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB40 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
 CB50 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
 CB60 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
 CB70 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
 CB80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBC0 1E 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CBF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM C4 D2 61 46 EA DE 4A FD FB F2 AA 49 38 F2 E5 3B 165

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CC00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC20 1E 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC40 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC50 83 C8 07 FF 00 00 00 00 00 00 00 00 00 00 00
 CC60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC70 3C 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CC80 FF FB 1C 01 E8 1C 03 0C 00 00 00 00 00 00 00 00
 CC90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CCA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CCB0 FF C8 07 00 FB 0E 00 00 00 00 00 00 00 00 00
 CCC0 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CCD0 8C 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CCE0 81 C8 01 00 00 00 00 00 00 00 00 00 00 00 00
 CCF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM E0 53 80 52 2E 6A 64 DC AD FB DC 40 6E D1 D7 3C 8B

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CD00 87 3E 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD50 FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD60 1C 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD70 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CD90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CDA0 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CDB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CDC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CDD0 83 F8 31 C7 18 7C 18 38 00 00 00 00 00 00 00
 CDE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CDF0 1E 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 2E 9D 6E E8 D6 16 25 FD 2D 5D 18 EF 92 8A C8 82 195

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CE00 1F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CE10 80 3F FF 00 00 00 00 00 00 00 00 00 00 00 00
 CE20 8F 1F 18 38 00 00 00 00 00 00 00 00 00 00 00
 CE30 FF FB 08 18 04 00 00 00 00 00 00 00 00 00 00
 CE40 18 0C FF 18 C3 18 00 00 00 00 00 00 00 00 00
 CE50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CE60 C7 18 C5 54 1E 00 00 00 00 00 00 00 00 00 00
 CE70 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CE80 87 FC CC 00 00 00 00 00 00 00 00 00 00 00 00
 CE90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CEA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CEB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CEC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CED0 18 0C 8C 78 38 1F C8 00 00 00 00 00 00 00 00
 CEE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CEF0 1E 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 4A 44 D1 C2 CB 31 D6 26 BE E5 27 AD 7D 18 5A 73 192

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 CF00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF10 83 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF40 86 83 C8 00 00 00 00 00 00 00 00 00 00 00 00
 CF50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CF80 83 8C 84 00 18 04 00 00 00 00 00 00 00 00 00
 CF90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CFA0 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CFB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CFC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CFD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CFE0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
 CFF0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
 VSUM 8A 56 7A 8D C9 E3 E7 12 3B 2A 4A 6E 72 78 2F E9 1A3

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D010 1F 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D030 78 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D040 86 83 C8 00 00 00 00 00 00 00 00 00 00 00 00
 D050 F8 06 83 8C 00 00 00 00 00 00 00 00 00 00 00
 D060 08 33 8C 00 00 00 00 00 00 00 00 00 00 00 00
 D070 83 8C 00 00 00 00 00 00 00 00 00 00 00 00 00
 D080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D0A0 7F FC 00 83 8C 18 7F C8 00 00 00 00 00 00 00
 D0B0 80 83 FF FF 00 00 00 00 00 00 00 00 00 00 00
 D0C0 83 71 FC 0E 81 84 3C 01 FC E4 87 FC 84 0F 18 84 32
 D0D0 38 F8 0C C3 18 1C 00 00 00 00 00 00 00 00 00
 D0E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D0F0 38 83 88 7C FF FC FC 00 83 83 FF FF 38 18 0E 88
 VSUM AF D0 49 3A A6 41 C8 86 94 88 3D AF E1 A7 73 45 18C

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D100 38 78 07 31 C8 0F FC C8 08 38 08 8F FC 38 38 1E 4
 D110 18 40 07 FC 00 00 00 00 00 00 00 00 00 00 00 00
 D120 8E 07 0F 3E 7F FF E5 01 C8 CF FF 08 81 DC 0F 3F
 D130 FF C7 0E 81 C8 0F FC 0E 81 C8 0F FC 0F 83 83 86
 D140 00 00 01 E2 00 00 00 16 00 00 18 0F FA 08 08 16
 D150 18 1F FE 10 00 28 12 3F C1 C8 2F FB 18 20 10 32
 D160 31 A8 18 00 00 00 00 00 00 00 00 00 00 00 00 00
 D170 78 80 00 00 78 00 1E 00 00 00 00 00 00 00 00 00
 D180 38 28 78 00 21 08 82 22 0E 3F FF F8 00 00 00 DF
 D190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D1A0 8A 00 00 18 C8 0C 18 00 00 00 00 00 00 00 00 00
 D1B0 80 83 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D1C0 7F FC 00 83 8C 18 7F C8 00 83 8C 0F FC 00 83 8C
 D1D0 84 83 FF 00 00 00 00 00 00 00 00 00 00 00 00 00
 D1E0 3F FF C8 00 00 00 00 00 00 00 00 00 00 00 00 00
 D1F0 FF 83 81 18 88 01 FF 00 00 00 00 00 00 00 00 00
 VSUM BB D5 46 D8 4F C9 54 56 83 39 D0 F4 A3 88 C8 132

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D200 18 1C 00 00 00 00 00 00 00 00 00 00 00 00 00
 D210 78 06 12 D0 FF 18 10 00 18 10 00 00 20 00 00 C9
 D220 5C 00 01 87 00 00 00 78 00 00 1E 87 FF 00 28 16
 D230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D240 01 C1 C8 0F FF F8 10 00 00 00 00 00 00 00 1F DD
 D250 FF F8 00 28 00 00 DF F8 01 00 85 87 1F 18 3C 18 1F
 D260 00 00 01 F3 00 00 00 00 00 00 1F FE 48 C1 00 44 FE 4E
 D270 44 48 02 44 FE 4E 44 02 44 FE 4E 44 02 8C 00 00
 D280 4F FE 14 C1 1E 81 00 00 00 00 00 33 FC 38 33 AD
 D290 FC C2 00 00 3E 32 00 3F 33 FC 33 83 FC 31 7F 38
 D2A0 38 F2 00 38 73 FC 38 33 FC 00 00 00 00 00 00 00
 D2B0 27 FC 38 47 FC 00 00 00 00 00 00 00 00 00 00 00
 D2C0 48 00 00 00 18 C8 48 28 00 00 00 00 00 00 00 00
 D2D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D2E0 86 18 61 85 06 61 87 CB 01 85 68 61 86 38 1E 06 97
 D2F0 1C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 2C 53 55 E0 4A 1B 08 87 07 4B EF 79 3F 47 3A BF 1E9

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D300 92 00 FC 92 01 80 92 03 92 03 92 03 92 03 92 03
 D310 53 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D320 F8 06 31 C8 03 33 00 01 3C 00 7F FF F8 00 00 F8 C
 D330 3F FF F8 00 00 00 00 00 00 00 00 00 00 00 00 00
 D340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D360 83 FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00
 D370 28 83 FF 00 00 00 00 00 00 00 00 00 00 00 00 00
 D380 84 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D390 3C 38 00 00 7F 83 8F 84 3F 87 C7 C8 00 61 E3 68 18
 D3A0 83 F8 71 00 00 00 00 00 00 00 00 00 00 00 00 00
 D3B0 28 83 FF F8 00 00 00 00 00 00 00 00 00 00 00 00
 D3C0 84 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D3D0 83 C7 F8 00 00 00 00 00 00 00 00 00 00 00 00 00
 D3E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D3F0 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 1A 74 96 42 00 AF EC 48 FB BD FC 87 AF 14 36 116

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D4F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM 8A 56 7A 8D C9 E3 E7 12 3B 2A 4A 6E 72 78 2F E9 1A3

ADDR +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :SUM
 D500 81 E1 05 7F 7E 7F 38 16 16 12 11 58 E8 21 0C 3C 1
 D510 05 C5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C5
 D520 8E 50 38 01 24 6F 18 F5 D3 5F C3 43 C7 21 00 00 5F
 D530 39 D3 5C 31 88 0F 11 00 00 00 00 00 00 00 00 00
 D540 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D550 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D560 31 88 0E 86 05 05 05 05 05 05 05 05 05 05 05
 D570 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D580 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D590 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D5A0 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D5B0 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D5C0 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D5D0 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
 D5E0 28 D0 C2 5F 9F 0C 8C 5D 7F 96 84 C5 ED AF 82
 D5F0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D600 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D610 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D620 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D630 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D640 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D650 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D660 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D670 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D680 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D690 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6A0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6B0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6C0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6D0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6E0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 D6F0 ED AD ED AD ED AD 7B C5 4C 38 01 14 5F ED AD ED 152
 VSUM 91 CA 31 58 DF 00 BE BE DA 47 00 C7 F9 AD C2 5E 1E5

リストD

C400	A	1	+2	3	4	5	6	7	8	A	+B	C	D	E	+F	14M																																																																																																			
C402	A	2	3	4	7	8	9	10	11	B	+6	G	H	I	23	72	C	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
C430	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100														
C438	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100														
C440	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100														
C448	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100														
C450	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48																																																																		

```
VSUM 5D B5 AC 0A 3D C9 08 A3 DB 7C 50 9F 95 01 04 42 :1B
```

CDDR	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+G	+H	+I	+J	+K	+L	+M	+N	+O	+P	+Q	+R	+S	+T	+U	+V	+W	+X	+Y	+Z	+AA	+AB	+AC	+AD	+AE	+AF	+AG	+AH	+AI	+AJ	+AK	+AL	+AM	+AN	+AO	+AP	+AQ	+AR	+AS	+AT	+AU	+AV	+AW	+AX	+AY	+AZ	+BA	+BB	+BC	+BD	+BE	+BF	+BG	+BH	+BI	+BJ	+BK	+BL	+BM	+BN	+BO	+BP	+BQ	+BR	+BS	+BT	+BU	+BV	+BW	+BX	+BY	+BZ	+CA	+CB	+CC	+CD	+CE	+CF	+CG	+CH	+CI	+CJ	+CK	+CL	+CM	+CN	+CO	+CP	+CQ	+CR	+CS	+CT	+CU	+CV	+CW	+CX	+CY	+CZ	+DA	+DB	+DC	+DD	+DE	+DF	+DG	+DH	+DI	+DJ	+DK	+DL	+DM	+DN	+DO	+DP	+DQ	+DR	+DS	+DT	+DU	+DV	+DW	+DX	+DY	+DZ	+EA	+EB	+EC	+ED	+EE	+EF	+EG	+EH	+EI	+EJ	+EK	+EL	+EM	+EN	+EO	+EP	+EQ	+ER	+ES	+ET	+EU	+EV	+EW	+EX	+EY	+EZ	+FA	+FB	+FC	+FD	+FE	+FF	+FG	+FH	+FI	+FJ	+FK	+FL	+FM	+FN	+FO	+FP	+FQ	+FR	+FS	+FT	+FU	+FV	+FW	+FX	+FY	+FZ	+GA	+GB	+GC	+GD	+GE	+GF	+GG	+GH	+GI	+GJ	+GK	+GL	+GM	+GN	+GO	+GP	+GQ	+GR	+GS	+GT	+GU	+GV	+GW	+GX	+GY	+GZ	+HA	+HB	+HC	+HD	+HE	+HF	+HG	+HH	+HI	+HJ	+HK	+HL	+HM	+HN	+HO	+HP	+HQ	+HR	+HS	+HT	+HU	+HV	+HW	+HX	+HY	+HZ	+IA	+IB	+IC	+ID	+IE	+IF	+IG	+IH	+II	+IJ	+IK	+IL	+IM	+IN	+IO	+IP	+IQ	+IR	+IS	+IT	+IU	+IV	+IW	+IX	+IY	+IZ	+JA	+JB	+JC	+JD	+JE	+JF	+JG	+JH	+JI	+JJ	+JK	+JL	+JM	+JN	+JO	+JP	+JQ	+JR	+JS	+JT	+JU	+JV	+JW	+JX	+JY	+JZ	+KA	+KB	+KC	+KD	+KE	+KF	+KG	+KH	+KI	+KJ	+KK	+KL	+KM	+KN	+KO	+KP	+KQ	+KR	+KS	+KT	+KU	+KV	+KW	+KX	+KY	+KZ	+LA	+LB	+LC	+LD	+LE	+LF	+LG	+LH	+LI	+LJ	+LK	+LM	+LN	+LO	+LP	+LQ	+LR	+LS	+LT	+LU	+LV	+LW	+LX	+LY	+LZ	+MA	+MB	+MC	+MD	+ME	+MF	+MG	+MH	+MI	+MJ	+MK	+ML	+MN	+MO	+MP	+MQ	+MR	+MS	+MT	+MU	+MV	+MW	+MX	+MY	+MZ	+NA	+NB	+NC	+ND	+NE	+NF	+NG	+NH	+NI	+NJ	+NK	+NL	+NM	+NO	+NP	+NQ	+NR	+NS	+NT	+NU	+NV	+NW	+NX	+NY	+NZ	+OA	+OB	+OC	+OD	+OE	+OF	+OG	+OH	+OI	+OJ	+OK	+OL	+OM	+ON	+OO	+OP	+OQ	+OR	+OS	+OT	+OU	+OV	+OW	+OX	+OY	+OZ	+PA	+PB	+PC	+PD	+PE	+PF	+PG	+PH	+PI	+PJ	+PK	+PL	+PM	+PN	+PO	+PP	+PQ	+PR	+PS	+PT	+PU	+PV	+PW	+PX	+PY	+PZ	+QA	+QB	+QC	+QD	+QE	+QF	+QG	+QH	+QI	+QJ	+QK	+QL	+QM	+QN	+QO	+QP	+QQ	+QR	+QS	+QT	+QU	+QV	+QW	+QX	+QY	+QZ	+RA	+RB	+RC	+RD	+RE	+RF	+RG	+RH	+RI	+RJ	+RK	+RL	+RM	+RN	+RO	+RP	+RQ	+RR	+RS	+RT	+RU	+RV	+RW	+RX	+RY	+RZ	+SA	+SB	+SC	+SD	+SE	+SF	+SG	+SH	+SI	+SJ	+SK	+SL	+SM	+SN	+SO	+SP	+SQ	+SR	+SS	+ST	+SU	+SV	+SW	+SX	+SY	+SZ	+TA	+TB	+TC	+TD	+TE	+TF	+TG	+TH	+TI	+TJ	+TK	+TL	+TM	+TN	+TO	+TP	+TQ	+TR	+TS	+TT	+TU	+TV	+TW	+TX	+TY	+TZ	+UA	+UB	+UC	+UD	+UE	+UF	+UG	+UH	+UI	+UJ	+UK	+UL	+UM	+UN	+UO	+UP	+UQ	+UR	+US	+UT	+UU	+UV	+UW	+UX	+UY	+UZ	+VA	+VB	+VC	+VD	+VE	+VF	+VG	+VH	+VI	+VJ	+VK	+VL	+VM	+VN	+VO	+VP	+VQ	+VR	+VS	+VT	+VU	+VV	+VW	+VX	+VY	+VZ	+WA	+WB	+WC	+WD	+WE	+WF	+WG	+WH	+WI	+WJ	+WK	+WL	+WM	+WN	+WO	+WP	+WQ	+WR	+WS	+WT	+WU	+WV	+WW	+WX	+WY	+WZ	+XA	+XB	+XC	+XD	+XE	+XF	+XG	+XH	+XI	+XJ	+XK	+XL	+XM	+XN	+XO	+XP	+XQ	+XR	+XS	+XT	+XU	+XV	+XW	+XX	+XY	+XZ	+YA	+YB	+YC	+YD	+YE	+YF	+YG	+YH	+
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---

VSUM 3A 8F DA F3 F1 9D 27 F8 EF FB 58 CE 32 84 F1 90 :7A

[illegible]

V30W 7D FC 16 BB 05 A0 D9 8E JA F8 C1 9F 38 67 19 35 :D7

[illegible]

VSUM 67 1B ED 3A 74 3F 65 D6 DC 31 0F 46 D3 FA EE BC :70

C008	F1	12	F3	14	F5	16	F7	18	F9	1C	F11	20	F13	22	F15	24	F17	26	F19	28	F21	30	F23	32	F25	34	F27	36	F29	38	F31	40	F33	42	F35	44	F37	46	F39	48	F41	50	F43	52	F45	54	F47	56	F49	58	F51	60	F53	62	F55	64	F57	66	F59	68	F61	70	F63	72	F65	74	F67	76	F69	78	F71	80	F73	82	F75	84	F77	86	F79	88	F81	90	F83	92	F85	94	F87	96	F89	98	F91	100	F93	102	F95	104	F97	106	F99	108	F101	110	F103	112	F105	114	F107	116	F109	118	F111	120	F113	122	F115	124	F117	126	F119	128	F121	130	F123	132	F125	134	F127	136	F129	138	F131	140	F133	142	F135	144	F137	146	F139	148	F141	150	F143	152	F145	154	F147	156	F149	158	F151	160	F153	162	F155	164	F157	166	F159	168	F161	170	F163	172	F165	174	F167	176	F169	178	F171	180	F173	182	F175	184	F177	186	F179	188	F181	190	F183	192	F185	194	F187	196	F189	198	F191	200	F193	202	F195	204	F197	206	F199	208	F201	210	F203	212	F205	214	F207	216	F209	218	F211	220	F213	222	F215	224	F217	226	F219	228	F221	230	F223	232	F225	234	F227	236	F229	238	F231	240	F233	242	F235	244	F237	246	F239	248	F241	250	F243	252	F245	254	F247	256	F249	258	F251	260	F253	262	F255	264	F257	266	F259	268	F261	270	F263	272	F265	274	F267	276	F269	278	F271	280	F273	282	F275	284	F277	286	F279	288	F281	290	F283	292	F285	294	F287	296	F289	298	F291	300	F293	302	F295	304	F297	306	F299	308	F301	310	F303	312	F305	314	F307	316	F309	318	F311	320	F313	322	F315	324	F317	326	F319	328	F321	330	F323	332	F325	334	F327	336	F329	338	F331	340	F333	342	F335	344	F337	346	F339	348	F341	350	F343	352	F345	354	F347	356	F349	358	F351	360	F353	362	F355	364	F357	366	F359	368	F361	370	F363	372	F365	374	F367	376	F369	378	F371	380	F373	382	F375	384	F377	386	F379	388	F381	390	F383	392	F385	394	F387	396	F389	398	F391	400	F393	402	F395	404	F397	406	F399	408	F401	410	F403	412	F405	414	F407	416	F409	418	F411	420	F413	422	F415	424	F417	426	F419	428	F421	430	F423	432	F425	434	F427	436	F429	438	F431	440	F433	442	F435	444	F437	446	F439	448	F441	450	F443	452	F445	454	F447	456	F449	458	F451	460	F453	462	F455	464	F457	466	F459	468	F461	470	F463	472	F465	474	F467	476	F469	478	F471	480	F473	482	F475	484	F477	486	F479	488	F481	490	F483	492	F485	494	F487	496	F489	498	F491	500	F493	502	F495	504	F497	506	F499	508	F501	510	F503	512	F505	514	F507	516	F509	518	F511	520	F513	522	F515	524	F517	526	F519	528	F521	530
------	----	----	----	----	----	----	----	----	----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----

7300 10 4E 80 10 40 00 01 10 0E 00 00 20 01 00 2E 00 100

[illegible]

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CA2B CB 01 FE 18 BD 2C 7E 2C 2C 3B 87 2D 01 FE 1A7
 CA3B 18 0C C3 00 CB 00 00 00 00 00 00 00 00 00 00 A6
 CA4B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA5B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA6B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA7B 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
 CA8B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CA9B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CAF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CB10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 VSUM EG 04 C4 1B DB 34 84 84 0C 3B 8A 37 EB EC 04 14 :A3

リストE

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CB80 21 9B 98 36 FF 2C 7E 2F 49 21 9B 98 36 7E FE :37
 CB10 86 D2 6C CF 4D 0E 00 86 04 0F 3B 01 19 EB 29 8B :74
 CB20 47 3E 0C 90 21 00 86 04 0F 3B 01 19 EB 29 8B :74
 CB30 18 7F 29 29 7C D1 0E 00 86 04 0F 3B 01 19 EB 29 8B :74
 CB40 20 2D 2D 34 18 0E 00 86 04 0F 3B 01 19 EB 29 8B :74
 CB50 84 6F C5 2E 77 2C 7E 7E 2E 84 6F C3 C3 CB :E5
 CB60 7E 3D CD 47 CB 7E 2C 46 80 2D 77 3A 98 98 3C 32 :81
 CB70 9B 98 36 18 2D 01 FE 1A 7C 2C 7E 2C 7E 2C 7E :1A
 CB80 3B 8E 2C 2D 7E 7E 2E 84 6F C3 C3 CB :E5
 CB90 20 2C 7E 57 FE 2E 84 6F C3 C3 CB :E5
 CBA0 18 85 FE 8A 28 81 CD 77 2D 72 18 01 FE 1A 28 23 :D8
 CBB0 14 2C 7E 8A 28 81 CD 77 2D 72 18 01 FE 1A 28 23 :D8
 CBC0 FC 7E 2C 7E 57 FE 2E 84 6F C3 C3 CB :E5
 CBD0 72 18 AA FE 17 28 85 14 72 C3 CB 19 28 2D :63
 CBE0 2C 7E 7E 08 28 14 5F 12 2C 2C 36 81 2D 2D :78
 CBF0 CC 38 01 1C 73 2D 72 C3 CB FE 89 28 2D 3B :85
 VSUM 4B 52 FB 3D 15 64 29 77 69 3A 48 55 7B 69 3B 02 :44

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CC00 ES 2C 36 81 2D 2D 14 3D 5F 18 E7 14 3D 7E :4F
 CC10 5F 0C 2E 28 0A 14 7D CC 3B 8E 2C 2C 7E :4F
 CC20 2D 7E 14 38 0E 15 18 C5 15 18 C2 9C :AA
 CC30 CC 7E 57 2C 19 28 12 7E 7E 8C 28 14 3D 77 :8E
 CC40 2D 72 C3 7D CB 14 3D 5F 18 E7 14 3D 7E :4F
 CC50 8B 28 12 3D 5F 0C 7D 7C 2C 2C 36 82 2D :C6
 CC60 2D 73 C3 7D CB 14 3D 5F 18 E7 14 3D 7E :4F
 CC70 C3 7D CB 7E 8C 28 ED 51 20 80 86 87 05 28 :78
 CC80 83 01 47 C9 7E 8A 28 86 7D CB 18 F8 2C 7E :2E
 CC90 8B 28 86 7D CB 18 F8 2C 7E 8C 28 14 3D :78
 CCA0 7E 57 2C 7E 1A 28 1E 7E 86 28 8A 3D 2C :62
 CCB0 36 83 2D 2D C3 2A CD 3D 5F 0C 7D 7C 7B :8F
 CCC0 C3 7D CB FE 18 28 2C 36 83 2D 7E 87 :D9
 CCD0 28 83 2C 18 55 FE 8B 28 18 0E 82 5F 15 CD :7E
 CCE0 38 01 14 73 2D 72 C3 CB D6 82 15 18 3C :FE
 CCF0 87 28 86 2C 2C 36 83 2D 18 E7 06 83 77 C3 :7D
 VSUM 3D 82 45 47 D8 FE 5B 3C 14 5E 86 C3 EE 8D 88 77 :8A

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CD00 CB FE 03 C7 E7 CD 7E 2C 7E 2C 7E 2C 7E :2C
 CD10 2C 86 C4 C3 7D CB FE 1A 28 3B 2C 7E 83 28 :29
 CD20 7E 84 28 13 2C 36 84 2D 5F 0C 7D 7C 38 01 :C2
 CD30 1D 73 2D 72 C3 7D CB 3D 5F 0C 7D 7C 38 01 :C2
 CD40 28 87 3D 77 2D 72 C3 7D CB 3D 5F 0C 7D 7C 38 01 :C2
 CD50 36 83 2D 2D C3 2A CD 3D 5F 0C 7D 7C 38 01 :C2
 CD60 82 15 14 C3 3D 18 0E 00 86 04 0F 3B 01 19 :EB
 CD70 28 24 2C 7E 15 28 86 28 13 2C 36 85 2D 5F :85
 CD80 0C 78 CC 36 81 15 73 2D 72 C3 7D CB 2D 72 :2C
 CD90 2C 85 C3 7D CB 8E 28 2C 36 86 C3 7D CB FE :3A
 CDA0 15 2C 7E 2D 72 2C 36 82 7C 7D CB FE 83 28 :0F
 CDB0 36 85 C3 7D CB 8E 28 2C 36 86 C3 7D CB FE :3A
 CDC0 15 72 C3 7D CB 8E 28 2C 36 86 C3 7D CB FE :3A
 CDD0 72 7C 7D CB 8E 28 2C 36 86 C3 7D CB FE :3A
 CDE0 72 7C 7D CB 8E 28 2C 36 86 C3 7D CB FE :3A
 CDF0 38 05 14 7B C3 2A CD 7D C6 06 4F 4A FE 83 :36
 VSUM C6 7C 8C 1E D1 41 51 35 18 12 51 10 B1 77 EB :F8

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CE00 D0 7C 2A CD FE 85 28 7E 57 FE 83 28 18 2C :B9
 CE10 7E 15 C6 2C 7E 2C 2C 36 86 C3 7D CB FE 3A :5A
 CE20 84 28 26 2C 7E 7E 2E 28 5F 0A 28 7E 85 25 :36
 CE30 8D 15 C6 2C 7E 2C 2C 36 86 C3 7D CB FE 3A :5A
 CE40 CD 7C CD 0A CD 7E 2C 15 FE 83 28 07 :8A
 CE50 34 77 2D 72 C3 7D CB C3 7F CD FE 86 C2 CA :86
 CE60 7E 57 FE 8A 28 81 CD 77 2D 72 18 01 FE 1A :28
 CE70 2C 36 87 C3 7D CB FE 8A 28 14 14 2C 7E :2C
 CE80 8F C6 83 5F CD 7C 38 8E 15 73 2D 72 C3 7D :CB
 CE90 7E 2E 84 6F C3 C3 CB :E5
 CEA0 FE 28 14 C6 83 5F CD 7C 38 8E 15 73 2D :72
 CEB0 2C 2C 2C 36 88 0C 7D C6 83 5F C3 8B C2 7E :93
 CEC0 34 5F CD 7C 38 8E 15 73 2D 72 C3 7D :CB
 CED0 2C 14 14 3C 5F CD 7C 38 8E 15 73 2D :72
 CEE0 2D 73 2D 72 C3 7D CB 2C 7E 5F 1C 14 2C :2C
 CEF0 8D 2D 7D FE 87 28 86 2C 36 83 2D 36 :86
 VSUM EB 77 63 D7 5E A2 42 16 48 25 62 53 D9 BC 49 D4 :EF

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 CF00 00 2C 2C 3E 3E 3E 3E 3E 3E 3E 3E 3E 3E :3E
 CF10 BE 21 9B 98 7E C6 78 6F 7E 32 68 98 01 81 :82
 CF20 37 BE C3 7D CB FE 8A 28 3C 2A CD FE 82 2D :A7
 CF30 83 C3 EE CB FE 8A 28 3C 2A CD FE 82 2D :A7
 CF40 56 0F 4F 2C 2C 2C 7E 68 78 68 28 14 3A :78
 CF50 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 CF60 4F 01 80 28 0B 78 01 28 0B C3 71 C6 11 :8F
 CF70 47 98 01 81 81 C7 37 BE 0D 00 8B BE 36 30 :81
 CF80 84 16 47 E3 CD 37 BE 7B C6 85 5F 3A 48 :98
 CF90 CD 37 BE 3E 28 0E 01 CD 37 BE 3A 47 98 :01
 CFA0 37 BE 86 2D 21 A6 98 3A 65 98 47 28 0D :98
 CFB0 2C 7E 8A 2D 38 84 2C 72 2D 77 78 81 82 :86
 CFC0 E8 05 CD 37 BE 81 7E 87 87 87 87 87 87 :87
 CFD0 98 21 C8 BE 0D 21 A6 98 3A 65 98 47 28 :98
 CFE0 BE 08 25 08 D6 84 CB FB 8D C6 D4 86 24 :38
 CFF0 FB 25 C6 8A 6F 79 01 81 86 11 EF 85 CD :37
 VSUM F8 0D E2 7D 83 77 58 53 84 47 88 28 EC 58 8D :39

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 D000 01 7C BE 81 85 CD 37 BE E1 7D BE 81 CD :37
 D010 BE 81 85 CD 37 BE E1 7D BE 81 CD :37
 D020 BE 81 85 CD 37 BE E1 7D BE 81 CD :37
 D030 98 06 85 56 8A 28 24 2C 2C 2C 2C 2C 2C :2C
 D040 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D050 21 80 80 80 80 80 80 80 80 80 80 80 80 :80
 D060 D6 22 AC 98 E1 2C 7E CD 47 D0 29 29 85 :29
 D070 29 29 85 D1 E1 29 19 29 29 29 29 29 29 :29
 D080 00 80 3A AA 98 98 78 81 4F 19 38 81 8C :87
 D090 D0 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D0A0 28 81 8C 98 78 81 4F 19 38 81 8C :87
 D0B0 00 80 3A AA 98 98 78 81 4F 19 38 81 8C :87
 D0C0 D1 E5 21 81 98 81 E1 8C 00 11 18 27 0D :E5
 D0D0 21 82 98 78 E1 8C 00 11 18 27 0D :E5
 D0E0 98 78 E1 8C 00 11 18 27 0D :E5
 D0F0 E1 8C 00 11 18 27 0D :E5
 VSUM 97 BC 89 FA 97 97 28 7A 81 FA 68 21 34 18 :5D

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 D100 21 98 98 78 18 1F 06 88 98 08 91 32 AE 58 :E2
 D110 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D120 3A AE 98 18 E3 11 08 0C 81 82 81 3E 35 :CD
 D130 11 88 05 81 82 81 3E 35 CD 3E 35 CD 3E :35
 D140 CD C2 D1 2E 8D 73 2D 72 C3 7D CB 14 3D :5F
 D150 8B 73 2D 72 C3 7D CB 14 3D 5F 18 E7 14 :3D
 D160 D1 CD 37 BE 21 98 98 7E A7 28 86 81 1E :86
 D170 CD 37 BE E1 2C 7E 87 8E 81 E5 CD 1B E1 :2C
 D180 BE E1 2C 7E 87 8E 81 E5 CD 1B E1 2C 7E :1C
 D190 BE E1 2C 7E 87 8E 81 E5 CD 1B E1 2C 7E :1C
 D1A0 E1 2C 7E 87 8E 81 E5 CD 1B E1 2C 7E :1C
 D1B0 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D1C0 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D1D0 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D1E0 8A 38 FB 05 C6 8A 5F 7A 88 05 8A D6 8A :38
 D1F0 85 06 8A 5F 7A 88 05 8A D6 8A 38 FB :78
 VSUM 18 3D 01 A6 53 D9 28 E9 5E 38 FF FA 85 05 :4A

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 D200 14 7E 47 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D210 2E 18 7E 47 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D220 12 11 88 CA 01 81 81 3A 6F 98 A7 3E 28 :84
 D230 28 86 0C 37 BE 3E 3A 0E 81 CD 37 BE 21 :A8
 D240 15 18 CD 05 D1 3A 6F 98 A7 28 2E 21 :A4
 D250 98 3E 31 8F CD D2 7E 27 2A 84 1A 3C 12 :FE
 D260 41 D6 84 1D 12 7E 34 69 1A FE 8A 36 8E :07
 D270 98 98 36 89 2C 18 FB 98 36 89 2C 18 FB :98
 D280 8A CD 8F 02 7E 47 28 1A 21 B8 98 36 89 :28
 D290 4E 2D 1A 81 FE 8A 83 34 D6 1A 1D 1B 80 :28
 D2A0 5E 81 C3 21 08 8E 87 8F 8F 8F 8F 8F 8F :8F
 D2B0 1C 2C 18 F6 2E B8 8E 87 36 80 2C 18 FB :AF
 D2C0 98 18 3A 3E 18 0E 1E 86 3E AF CD E0 82 :6F
 D2D0 1E 86 3E 18 0E 1E 86 3E AF CD E0 82 :6F
 D2E0 87 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D2F0 38 8D 34 C6 8A 91 12 D1 F1 B8 28 EF 39 :86
 VSUM 8A 99 D8 64 8C BA EC C7 12 63 67 1E D8 8D :C4

ADDR +B +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +SUM
 D300 01 81 3A 6F 98 A7 28 82 86 82 3E 37 CD :61
 D310 11 7E 47 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D320 12 1E 14 2C 7E 87 87 87 87 87 87 87 87 :87
 D330 7E 87 87 87 87 87 87 87 87 87 87 87 87 :87
 D340 4A CB 27 CD 27 CD 27 CD 27 CD 27 CD 27 :CD
 D350 57 82 81 4F 2E AE 8E B8 28 3E 7E B8 28 :B8
 D360 21 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F 8F :8F
 D370 5C 83 FE 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F :3F
 D380 6E 12 FE 1E 38 82 1E 77 18 80 01 81 BF :4E
 D390 15 1E D6 38 82 D6 6F 46 45 74 48 FE 21 :6F
 D3A0 C3 88 CB 88 CB 88 CB 88 CB 88 CB 88 CB :88
 D3B0 28 83 FF F8 08 08 3F F8 08 08 08 08 08 :08
 D3C0 84 88 88 88 88 88 88 88 88 88 88 88 88 :88
 D3D0 83 C7 F8 88 88 88 88 88 88 88 88 88 88 :88
 D3E0 87 8E 08 08 08 08 08 08 08 08 08 08 08 :08
 D3F0 28 88 98 08 08 08 08 08 08 08 08 08 08 :08
 VSUM 58 FB 7A F3 7A 14 28 D9 83 D7 7E 8D 72 E1 :F8

住所録プログラム

MANDATA

住所から血液型まで個人情報データベースにできるプログラムです。あまり多くの人間は入りませんが、学校や会社の親しい友人やちょっと気になる人（？）のデータを入れるのには、ちょうど

どいい容量でしょう。プログラムはすべてBASICで書かれています。PC-8800シリーズ全機種対応になっていますが、VAやSR以降のマシンをお持ちの方は、漢字対応にすることも可能です。

佐藤 淳

プログラムの機能

このプログラムはいわゆる住所録というものに別目新しいものではありません。よく手帳の後ろなどについているアドレス帳のようなものと思っても結構です。私は普段あまり手帳やアドレス帳は持たず、必要なときはワープロで打ったものをプリンタで打ち出し、そのままの形で持っていることが多いので、いちいちワープロで出すのが面倒なのでこのプログラムを作りました。内容は本人の氏名・住所・電話番号・会社・学校名、生年月日など、早くいえば個人のデータベースです。入力は、漢字でできればベストだったのですが、私の88が一番初期のいわゆるPC-8801というもので、速度も遅く、漢字もうまく扱えないために、英数とカナ入力になってしまいました。SR以降の機種やVAをお持ちの人はご自分で漢字対応のものを作ってみてください。

プログラム自体はさほど難しくないとはいえますが、関数やサブルーチンを多用しているため、少し見にくくなってしまいました。

図1

***** Telephone & Address *****

*****	XXXXXXXXXXXXXXXXXXXX	No.	2
Name	ニッパ*ン タロウ		
Sex	Male		
Zip code	000-11		
Address	トウキョウト チヨダ*ク XX-XX- XX		
Telephone	03-000-0000		
Office	ホ*ウ シリツ タイガ*ク		
(Address)	サイタマ*ン トヨウ*シ XX-X -XXX		
Telephone	040-000-0000		
Position	イカ*ク		
Birth Day	1967年 6月23日		
Asterizm	カニ*ク		
Blood Type	B カ*タ		
MEMO			

対応機種

PC-8800シリーズ全機種 (VAを含む) で動きます。

- ① PC-8800シリーズ本体
- ② ディスク、プリンタは機種その他まったく関係なし

プログラムの入力

すべてBASICなので、入力は多少面倒かもしれませんが、変数名がかなり長くなってしまいましたが、面倒くささらずに入力してみてください。入力が終わったら、好きなファイル名でディスクにセーブしておいてください。

使用方法

プログラムを起動させると、

- ① NEW DATA
- ② OLD DATA

と聞いてきます。まだこの住所録に何もデータをセーブしていないときは、

NEW DATA

を、以前にデータをセーブしてあるときは、

OLD DATA

を選んでください。

NEW DATAを選択すると、

"Are you sure?"

と聞いてきます。

これは、もし①を選んでしまうと、以前残っていたデータがすべて消えてしまうからです。

このプログラムでは、住所録に登録できる人数は10名になっていますが、もっと人数を入りたい人は、プログラムの先頭にあるMNという変数を変えてください。

ただし8人で1クラス使うので、ディスクの容量と見比べて、人数を決めてください。なお、このプログラムでは2クラスをデータのファイルとして使っていますので、残りの容量が2クラス未満のときは別のディスクにセーブしてください。データファイルの確保ができると、1番目の人のデータが表示されます。右側に出ているのはコマンドリストです。まだ何もデータを入れていないときは、左側には表だけが表示されていないはずす。

次にコマンド説明を行います。

コマンドの説明

1) ROLL UP キー

このキーを押すと次のページが呼び出されます。一番最後のページでこのキーを押すと1ページ目が呼び出されます。

2) ROLL DOWN キー

このキーは前のROLL UP キーの逆で前のページを呼び出します。前のキーと同様、1ページ目でこのキーを押すと、一番最後のページが呼び出されるようになっていきます。

3) CTRL+I キー

このキーを押すとデータの入力を始めます。まず名前から入力のはじまります。項目をすべて英語にしてみました。わからない人のために、上から日本語訳をつけていきます。上から、名前、性別、郵便

番号, 住所, 電話番号, 会社・学校, その住所と電話番号, 部署もしくは学部, 誕生日, 星座, 血液型, 注釈となっています。性別と血液型は下に表示されるの中から番号を選んでください。

入力がすべて終わると, コマンド入力にもどります。

なお, 入力に際しては, INS と DEL キーは受けつけませんので注意してください。CLR キーを押すと, その項目の内容はすべて消えて, カーソルはその項目の一番前にきます。

また, 項目の1番先頭にいるとき DEL キーを押すと1つ前の項目にもどります。DEL キーが使えるのはこのときだけです。カーソルが一番右はじにくると自動的に次の項目に行ってしまうから, 間違えて RETURN キーを押してしまつたら, DEL キーを押して前の項目に移ってください。

4) CTRL+W キー

データをディスクにセーブします。これを行わないと, せっかく入力したデータがセーブされません, きちんとセーブしておきましょう。

5) CTRL+L キー

これは, 1ページ目からいきなり5ページ目に飛ぶときなどに使います。下に,

INPUT NUMBER

と聞いてくるので, そのページ数を入力してください。

6) CTRL+P キー

データをプリントアウトします。1回のプリントアウトで, 現在表示されているデータをプリンタに出力します。

7) CTRL+Q キー

プログラムが終了します。

リスト1

```

1000 *****
1010 *
1020 * TELEPHONE & ADDRESS NOTE *
1030 *
1040 * COPYRIGHT *
1050 *
1060 * Ver 1.1 1987/12/16 *
1070 *
1080 *****
1090
1100 WIDTH 80,25:CONSOLE 0,25,0,1
1110 DIM SUB$(40)
1120 DEFINT A-Z
1130 MN=10
1140 STOP ON
1150 ON STOP GOSUB *ST
1160 ON ERROR GOTO *BUG
1170 CLS
1180 PRINT "(1) NEW DATA"
1190 PRINT "(2) OLD DATA"
1200 AS=INKEY$
1210 IF AS="" THEN 1200
1220 AI=VAL(AS)
1230 IF AI<1 OR AI>2 THEN 1280
1240 IF AI=1 THEN 1260
1250 IF AI=2 THEN 1310
1260 PRINT "Are you sure?":
1270 AS=INPUT$(1):PRINT AS
1280 IF AS="" THEN 1260
1290 IF AS="Y" OR AS="y" OR AS="J" THEN GOSUB *MDATA
1300 IF AS="N" OR AS="n" OR AS="E" THEN 1170 ELSE 1270
1310 CLS:N=1
1320 GOTO *MAIN
1330 *ST
1340 CLOSE:CONSOLE 0,25:END
1350 *BUG
1360 IF ERR=55 THEN LOCATE 0,23:PRINT "data 0" :PRINT:RESUME NEXT
1370 ON ERROR GOTO 0
1380 *GRAPH
1390 CONSOLE 1,25
1400 CLS
1410 PRINT "***** Telephone & Address *****"
1420 PRINT
1430 PRINT "***** No. 000 *****"
1440 PRINT USING "##### No. 000 "N
1450 GOSUB *DOT
1460 PRINT USING "Name i & & "NAME$
1470 GOSUB *DOT
1480 PRINT USING "Sex i & & "SEX$
1490 GOSUB *DOT
1500 PRINT USING "Zip code i & & "ZIP$
1510 GOSUB *DOT
1520 PRINT USING "Address i & & "ADD1$
1530 GOSUB *DOT
1540 PRINT USING "i & & "ADD2$
1550 GOSUB *DOT
1560 PRINT USING "Telephone i & & "TEL$
1570 GOSUB *DOT
1580 PRINT USING "Office i & & "OFF1$
1590 GOSUB *DOT
1600 PRINT USING "(Address) i & & "OFFADD1$
1610 GOSUB *DOT
1620 PRINT USING "i & & "OFFADD2$
1630 GOSUB *DOT
1640 PRINT USING "Telephone i & & "OTEL$
1650 GOSUB *DOT
1660 PRINT USING "Position i & & "POS1$
1670 GOSUB *DOT
1680 PRINT USING "Birth Day i & & "BIRTH$
1690 GOSUB *DOT
1700 PRINT USING "Asterizm i & & "AST$
1710 GOSUB *DOT
1720 PRINT USING "Blood Type i & & "BLOOD$
1730 GOSUB *DOT
1740 PRINT USING "MEMO i & & "MEMO1$
1750 GOSUB *DOT
1760 PRINT USING "i & & "MEMO2$
1770 GOSUB *DOT
1780 PRINT
1790 RETURN
1800 *SHOWCOM
1810 X=3

```

```

1828 LOCATE 40,X
1830 PRINT "----- Command -----"
1840 LOCATE 40,X+2
1850 PRINT " (Roll up) - Page up
1860 LOCATE 40,X+4
1870 PRINT " (Roll down) - Page down
1880 LOCATE 40,X+6
1890 PRINT " (CTRL + I ) - Input DATA
1900 LOCATE 40,X+8
1910 PRINT " (CTRL + Q ) - Quit
1920 LOCATE 40,X+10
1930 PRINT " (CTRL + W ) - Save DATA
1940 LOCATE 40,X+12
1950 PRINT " (CTRL + L ) - Load DATA
1960 LOCATE 40,X+14
1970 PRINT " (CTRL + P ) - Print out DATA
1980 RETURN
1990 *MAIN
2000 GOSUB *OP
2010 GOSUB *RDATA
2020 GOSUB *GRAPH:GOSUB *SHOWCOM
2030 GOSUB *COMMAND
2040 *COMMAND
2050 LOCATE 40,20
2060 PRINT "Input Command ":
2070 AS=INPUT$(1)
2080 IF AS="" THEN 2070
2090 A=ASC(AS)
2100 IF A=8H8 THEN GOSUB *UP
2110 IF A=8HF9 THEN GOSUB *DOWN
2120 IF A=8B9 THEN GOSUB *INDATA
2130 IF A=8H11 THEN CLOSE:WIDTH 80,25:END
2140 IF A=8HC THEN GOSUB *LDATA
2150 IF A=8H17 THEN GOSUB *WDATA
2160 IF A=8H18 THEN GOSUB *PDATA
2170 GOTO 2050
2180 *OP
2190 OPEN "TELAD.DAT" AS #1
2200 FIELD #1,19 AS DNAME$,1 AS DSEI$,6 AS DZIP$,38 AS DADD$,15 AS DTEL$
2210 FIELD #1, 82 AS DUMMY$,19 AS DOFF$,38 AS DOFFADD$,15 AS DOTE$,19 AS DPOS
17$
2220 FIELD #1,173 AS DUMMY2$,16 AS DBIRTH$,17 AS DAST$,1 AS DBLD$,38 AS DMEMO$
2230 RETURN
2240 *DOT
2250 LOCATE 36,CSRLIN-1:PRINT "███"
2260 RETURN
2270 *RDATA
2280 SEX$=""
2290 BLOOD$=""
2300 GET #1,N
2310 K=N
2320 NAME$=DNAME$
2330 SEI=DSEI$
2340 IF VAL(SEI)=1 THEN SEX$="Male"
2350 IF VAL(SEI)=2 THEN SEX$="Female"
2360 ZIP=DZIP$
2370 ADD=DADD$
2380 ADD1=MID$(DADD$, 1,19)
2390 ADD2=MID$(DADD$,20,38)
2400 TEL=DTEL$
2410 OFF1=DOFF$
2420 OFFADD=DOFFADD$
2430 OTEL=DOTE$
2440 OFFADD1=MID$(DOFFADD$, 1,19)
2450 OFFADD2=MID$(DOFFADD$,20,19)
2460 POSIT=DPOSIT$
2470 BIRTH=DBIRTH$
2480 AST=DAST$
2490 BLD=DBLD$
2500 BLD=VAL(BLD$)
2510 IF BLD=1 THEN BLOOD$="A ♀♀"
2520 IF BLD=2 THEN BLOOD$="B ♀♀"
2530 IF BLD=3 THEN BLOOD$="O ♀♀"
2540 IF BLD=4 THEN BLOOD$="AB ♀♀"
2550 MEMO=DMEMO$
2560 MEMO1=MID$(DMEMO$, 1,19)
2570 MEMO2=MID$(DMEMO$,20,19)
2580 RETURN
2590 *INDATA
2600 XD=15:YD=5:L=19 'name
2610 GOSUB *EDSUB:NAME$=INN$
2620 IF SSW=1 THEN LOCATE X,Y:GOTO 2610
2630 IF SSW=2 THEN 3140
2640 XD=15:YD=6:L=6 'Sex
2650 GOSUB *EDSUB2
2660 IF SSW=1 THEN GOTO 2680
2670 IF SSW=2 THEN 3140
2680 XD=15:YD=7:L=6 'Zip code
2690 GOSUB *EDSUB3:ZIP$=INN$
2700 IF SSW=1 THEN GOTO 2640
2710 IF SSW=2 THEN 3140
2720 XD=15:YD=8:L=40 'Address
2730 GOSUB *EDSUB:ADD$=INN$
2740 ADD1=MID$(ADD$,1,19):ADD2=MID$(ADD$,20,19)
2750 IF SSW=1 THEN GOTO 2680
2760 IF SSW=2 THEN 3140
2770 XD=17:YD=10:L=15 'Tel
2780 GOSUB *EDSUB4:TEL$=INN$
2790 IF SSW=1 THEN GOTO 2720
2800 IF SSW=2 THEN 3140
2810 XD=15:YD=11:L=19 'Office
2820 GOSUB *EDSUB:OFF1$=INN$
2830 IF SSW=1 THEN GOTO 2770
2840 IF SSW=2 THEN 3140
2850 XD=15:YD=12:L=40 '(Add)
2860 GOSUB *EDSUB:OFFADD$=INN$
2870 OFFADD1=MID$(OFFADD$,1,19):OFFADD2=MID$(OFFADD$,20,19)
2880 IF SSW=1 THEN GOTO 2810
2890 IF SSW=2 THEN 3140
2900 XD=17:YD=14:L=15 '(Tel)
2910 GOSUB *EDSUB4:OTEL$=INN$
2920 IF SSW=1 THEN GOTO 2850
2930 IF SSW=2 THEN 3140
2940 XD=15:YD=15:L=19 'Pos
2950 GOSUB *EDSUB:POSIT$=INN$
2960 IF SSW=1 THEN GOTO 2900
2970 IF SSW=2 THEN 3140
2980 XD=17:YD=16:L=16 'Birth day
2990 GOSUB *EDSUB5:BIRTH$=INN$
3000 IF SSW=1 THEN GOTO 2940

```

```

3810 IF SSW=2 THEN 3140
3820 XD=15:YD=17:L=17
3830 GOSUB *EDSUB:AST$=INN$
3840 IF SSW=1 THEN GOTO 2980
3850 IF SSW=2 THEN 3140
3860 XD=15:YD=18:L=8
3870 GOSUB *EDSUB
3880 IF SSW=1 THEN GOTO 3820
3890 IF SSW=2 THEN 3140
3900 XD=15:YD=19:L=48
3910 GOSUB *EDSUB:MEMO$=INN$
3920 MEMO1$=MID$(MEMO$,1,19):MEMO2$=MID$(MEMO$,20,19)
3930 IF SSW=1 THEN GOTO 3860
3940 RETURN
3950 *EDSUB
3960 X=XD:Y=YD:INL=L:SW=0:SSW=0:LN=1:INN$=""
3970 FOR I=1 TO L:SUB$(I)=*":NEXT
3980 IF L>19 THEN LN=19
3990 LOCATE X,Y:PRINT SPACE$(INL)
4000 IF L>19 THEN LOCATE X,Y+1:PRINT SPACE$(INL)
4010 LOCATE X,Y
4020 AS=INPUT$(1)
4030 IF AS="" THEN 3220
4040 AI=ASC(AS)
4050 PF L>19 AND AI=8HID AND SW=1 AND X<XD THEN X=XD+INL-1:Y=YD:SW=0:LN=LN-1
4060 IF L>19 AND AI=8HIC AND SW=0 AND X<XD+INL-1 THEN X=XD+YD+1:SW=1:LN=LN+1:GOTO 3480
4070 IF AI=8HD THEN X=X+1:LN=LN-1:IF X<XD THEN X=XD:LN=1:GOTO 3480 ELSE 3480
4080 IF AI=8HF AND X<XD THEN SSW=1:X=XD:GOTO 3410
4090 IF AI=8HIC THEN X=X+1:LN=LN+1:IF X<XD+INL-1 THEN 3410 ELSE 3480
4100 IF AI=8HC AND L<38 THEN X=XD:Y=YD:LN=1:LOCATE X,Y:PRINT SPACE$(INL):FOR I=1 TO L:SUB$(I)=*":NEXT:GOTO 3480
4110 IF AI=8HD THEN SW=0:SSW=0:GOTO 3410
4120 IF AI=8HF THEN 3220
4130 IF AI=8HID THEN SSW=2:GOTO 3410
4140 IF 8HIFAI AND 8H72AI THEN 3360 ELSE 3480
4150 PRINT AS:SUB$(LN)=AS
4160 X=X+1:LN=LN+1
4170 IF L>19 AND X<XD+INL-1 AND SW=0 THEN X=XD:Y=YD+1:SW=1:LN=LN
4180 IF X<XD+INL-1 THEN 3410
4190 GOTO 3210
4200 FOR I=1 TO L:INN$=INN$+SUB$(I):NEXT
4210 IF INN$="" THEN INN$=SPACE$(INL)
4220 RETURN
4230 *EDSUB
4240 X=XD:Y=YD:INL=L:SSW=0
4250 LOCATE 0,23:PRINT "(1) Male (2) Female --- Push 1 or 2"
4260 LOCATE X,Y
4270 AS=INPUT$(1)
4280 IF AS="" THEN 3480
4290 AI=ASC(AS)
4300 IF AI=8H7F THEN SSW=1:GOTO 3570
4310 IF AI=49 OR AI=8HD THEN SEX$="Male"
4320 IF AI=58 THEN SEX$="Female"
4330 IF AI=49 OR AI=58 THEN 3480
4340 LOCATE XD,YD:PRINT SEX$
4350 LOCATE 0,23:PRINT SPACE$(48)
4360 RETURN
4370 *EDSUB
4380 L=3:XD=XD:YD=YD:SSW=0
4390 LOCATE XD,3,YD
4400 PRINT "-"
4410 GOSUB *EDSUB
4420 IF SSW=1 THEN 3720
4430 INN1$=INN$
4440 L=2:XD=XD+4:YD=YD
4450 GOSUB *EDSUB
4460 IF SSW=1 THEN XD=XD-4:YD=YD:L=3:GOTO 3620
4470 INN2$=INN$
4480 IF INN1$="" THEN INN1$=SPACE$(3)
4490 IF INN2$="" THEN INN2$=SPACE$(2)
4500 INN$=INN1$+"-"+INN2$
4510 RETURN
4520 *EDSUB
4530 L=5:XD=XD:YD=YD:SSW=0
4540 LOCATE XD,5,YD
4550 PRINT "-"
4560 LOCATE XD,9,YD
4570 PRINT "-"
4580 GOSUB *EDSUB
4590 IF SSW=1 THEN 4880
4600 P=INSTR(INN$,"-")-1
4610 IF P=0 THEN P$=LEFT$(INN$,P) ELSE P$=INN$
4620 INN1$=RIGHT$(P$,"-P$,5)
4630 L=3:XD=XD+6:YD=YD
4640 GOSUB *EDSUB
4650 IF SSW=1 THEN XD=XD-6:YD=YD:L=5:GOTO 3790
4660 P=INSTR(INN$,"-")-1
4670 IF P=0 THEN P$=LEFT$(INN$,P) ELSE P$=INN$
4680 INN2$=RIGHT$(P$,"-P$,3)
4690 L=4:XD=XD+4:YD=YD
4700 GOSUB *EDSUB
4710 IF SSW=1 THEN XD=XD-4:YD=YD:L=3:GOTO 3850
4720 P=INSTR(INN$,"-")-1
4730 IF P=0 THEN P$=LEFT$(INN$,P) ELSE P$=INN$
4740 INN3$=RIGHT$(P$,"-P$,4)
4750 IF INN1$="" THEN INN1$=SPACE$(5)
4760 IF INN2$="" THEN INN2$=SPACE$(3)
4770 IF INN3$="" THEN INN3$=SPACE$(4)
4780 INN$=INN1$+"-"+INN2$+"-"+INN3$
4790 RETURN
4800 *EDSUB
4810 L=4:XD=XD:YD=YD:SSW=0
4820 LOCATE XD,4,YD
4830 PRINT "E"
4840 LOCATE XD,7,YD
4850 PRINT "R"
4860 LOCATE XD,10,YD
4870 PRINT "B"
4880 GOSUB *EDSUB
4890 IF SSW=1 THEN 4880
4900 INN1$=RIGHT$(STR$(VAL(INN$)),4)
4910 L=2:XD=XD+5:YD=YD
4920 GOSUB *EDSUB
4930 IF SSW=1 THEN XD=XD-5:YD=YD:L=4:GOTO 4890
4940 INN2$=RIGHT$(STR$(VAL(INN$)),2)
4950 L=2:XD=XD+3:YD=YD

```



```

4178 GOSUB *EDSUB
4188 IF SSN1 THEN XD=XD-3:YD=YD-1:L=2:GOTO 4138
4198 INN3=RIGHT$( " " +STR$(VAL(INN3)),2)
4208 IF INN1=0 THEN INN1=SPACES(4)
4218 IF INN2=0 THEN INN2=SPACES(2)
4228 IF INN3=0 THEN INN3=SPACES(2)
4238 INN=INN1$+" "+INN2$+" "+INN3$+" "
4248 RETURN
4258 *EDSUB
4268 X=XD:Y=YD:INL=L:SSW=0
4278 LOCATE 0,23:PRINT "(1) Aa* (2) Bb* (3) Cc* (4) Aa* --- PUSH 1-4"
4288 LOCATE X,Y
4298 AS=INPUT$(1)
4308 IF AS="" THEN GOTO 4298
4318 A1=ASC(AS)
4328 IF A1=47 THEN SSW=1:GOTO 4400
4338 IF A1=49 OR A1=51 THEN BLOOD$="A" カ*":BLD$="1":GOTO 4388
4348 IF A1=50 THEN BLOOD$="B" カ*":BLD$="2":GOTO 4388
4358 IF A1=51 THEN BLOOD$="O" カ*":BLD$="3":GOTO 4388
4368 IF A1=52 THEN BLOOD$="AB" カ*":BLD$="4":GOTO 4388
4378 IF A1<49 OR A1>52 THEN GOTO 4298
4388 LOCATE XD,YD:PRINT BLOOD$
4398 LOCATE 0,23:PRINT SPACES(58)
4400 RETURN
4410 *WDATA
4420 LOCATE 0,23:PRINT "Now Saving....."
4430 LSET DNAMES=NAMES
4440 LSET DSEX$=SEX$
4450 LSET DZIP$=ZIP$
4460 LSET DADD$=ADD$
4470 LSET DTEL$=TEL$
4480 LSET DOFF$=OFF$
4490 LSET DCOFFADD$=COFFADD$
4500 LSET DTEL$=TEL$
4510 LSET DPOSIT$=POSIT$
4520 LSET DBIRTH$=BIRTH$
4530 LSET DAST$=AST$
4540 LSET DBLOOD$=BLOOD$
4550 LSET DMEMO$=MEMO$
4560 PUT#1,N
4578 LOCATE 0,23:PRINT SPACES(48)
4588 RETURN
4598 *LDATA
4608 LOCATE 0,23
4618 PRINT "INPUT NUMBER:"
4628 XD=14:YD=23:L=3
4638 GOSUB *EDSUB
4648 A1=VAL(INN3)
4658 IF A1<1 OR A1>LOF(1) THEN GOTO 4628
4668 N=A1
4678 GOSUB *WDATA
4688 GOSUB *GRAPH
4698 GOSUB *SHOWCOM
4708 RETURN
4718 *UP
4728 N=N+1
4738 IF N>LOF(1) THEN N=1
4748 GOSUB *WDATA
4758 GOSUB *GRAPH
4768 GOSUB *SHOWCOM
4778 RETURN
4788 *DOWN
4798 N=N-1
4808 IF N<1 THEN N=LOF(1)
4818 GOSUB *WDATA
4828 GOSUB *GRAPH
4838 GOSUB *SHOWCOM
4848 RETURN
4858 *WDATA
4868 LPRINT
4878 LPRINT
4888 LPRINT
4898 LPRINT USING "***** Telephone & Address *****"
4908 LPRINT USING "##### No. ###" :N
4918 LPRINT USING "Name I & & :NAMES
4928 LPRINT USING "Sex I & & :SEX$
4938 LPRINT USING "Zip code I & & :ZIP$
4948 LPRINT USING "Address I & & :ADD1$
4958 LPRINT USING "Telephone I & & :ADD2$
4968 LPRINT USING "Office I & & :TEL$
4978 LPRINT USING "(Address) I & & :OFF1$
4988 LPRINT USING "I & & :COFFADD1$
4998 LPRINT USING "Telephone I & & :COFFADD2$
5008 LPRINT USING "Position I & & :OTEL$
5018 LPRINT USING "Birth Day I & & :POSIT$
5028 LPRINT USING "Asteriza I & & :BIRTH$
5038 LPRINT USING "Blood Type I & & :AST$
5048 LPRINT USING "MEMO I & & :BLOOD$
5058 LPRINT USING "MEMO I & & :MEMO1$
5068 LPRINT USING "MEMO I & & :MEMO2$
5078 RETURN
5088 *MDATA
5098 GOSUB *OP
5108 FOR I=1 TO MN
5118 N=I
5128 GOSUB *WDATA
5138 NEXT I:GOTO 5118
5148 RETURN 2010

```



竹田津 恩

このプログラムは、本誌の姉妹誌「Oh! 98」に掲載されたものをPC-88VA用に移植したものです。

この筆者は今回PC-88VAに初めて触れ、そのグラフィックス表示能力の素晴らしいさに驚かされたそうです。ここでは誰もが小学生のころによくやった「鉛筆飛ばし」をできるだけ忠実に再現しています。童心にかえってやってみてはいかがでしょうか？

懐かしの鉛筆飛ばし

PENCIL WARS⁸⁸VA版

うれし懐かし鉛筆飛ばし

私は「面倒臭がり」ですが、戦争ゲームは大好きです。最近では「大戦略」などのシミュレーションゲームが花盛りですが、私の幼少時代にはそんなエエモンはございませんでした。

中学生のころ、「ボードシミュレーションゲームをやろう」といわれてOKしたものの、分厚いルールブックを渡された上、「じゃあこれを明日までに覚えてきてね」といわれてまづオロオロ。しかもゲームを始めると、やれ攻撃値と防御値の計算だの、地形による補正だのと、その面倒なことといったらございません。たかがサイコロふって勝ち負け決めるのに、延々とその手の作業を繰り返すにはウンザリで、「もう2度とこんなゲームはやらないもんね」と固く心に誓ったものでした。

最近ではパソコンのウォーゲームにハマっていて、面倒な計算は機械がやってくれますが、それでも面倒臭いことは嫌いです。

そう思っているうちに思い出したのが「鉛筆飛ばし」。私の小学生のころは、必修科目でした。休み時間にノートを破って飛行機だけの戦車だのを書き込み、「今度は飛行機を動かすからな」などといいつつ鉛筆をはじいたものです。なにしろ今のうちにファミコンなんてないご時世のことですので、「これはあたる」「いや、あたるでない」「よし、ならいのちかけかあ！」と熱くなって遊んでいたのを覚えています。

世の中はちょうどレトロブーム。「鉛筆飛ばし」を、88で再現したのが、この「PENCIL WARS」です。

(注意…「鉛筆飛ばし」というのは私が育

ったところの呼び方ですので、ほかにもいろいろ呼び方があると思います)

プログラムの入力と実行

プログラムはBASIC(リスト1)だけです。

BASICプログラムは、BASICのエディタかPCエディタで入力してください。

プログラムの入力後は、
SAVE "PENCIL"

で必ずセーブしてください。

必要なハードウェア

① PC-88VA

② マウス(NEC純正品、もしくはその相当品)。また、プログラムの実行には、N88-日本語 BASIC V3が必要です。

ゲームを始める前に

このゲームは、すべてマウスで操作します。キーボードはプログラムの起動以外にはいっさい使いませんので、マウスを動かすスペースを取るために机の隅に寄せておいたほうがいでしょう(その理由はあとでわかります)。また、マウスのボタンは左側しか使いません。

遊び方

人間(プレイヤー)対コンピュータの空中戦です。それぞれ20機ずつの飛行機を持っていて、先に相手の飛行機を全滅させたほうが勝ちです。

画面の説明

プログラムをスタートさせると、画面に矢印のマウスカーソルが表示されます。手元のマウスを動かして、矢印が動くのを確

認しましょう。

一番上に表示されているのが、プレイヤーとコンピュータの飛行機の数です。どちらかの機数が0になったらゲームエンドです。

画面中央に、「PENCIL WARS」と書かれたウィンドウがあります。これが「鉛筆飛ばし」の戦場です。このゲームは、昔からノートの切れはして遊ぶものと相場が決まっていますので、その雰囲気をも再現しました。

ノートに「F-16」とおぼしき飛行機が表示されています。これがプレイヤーの飛行機で、ゲーム開始時にはバラバラに配置されています。コンピュータの飛行機はプレイヤーと同じ飛行機がひっくり返っているのですが、今はまだ見えません。

ノートの下に[移動][攻撃][終了]と表示されているのがコントロールパネルです。

最初は、[移動]になっています。

ノートのスクロール

ノートの穴(一番下の穴)がひとつだけ黒丸になっています。じつはノートの穴をマウスでクリック(マウスのボタンを押す)するとノートがスクロールします。試しに一番上の穴に矢印の先を合わせて、マウスの左側のボタンを押してください。ノートにコンピュータ側の飛行機が表示されます。

実際の戦場は、縦の長さは画面に表示されている長さの2倍あります(横幅は同じです)。接近戦になったら、一番見やすい位置に調節してください。

コントロールパネル

プレイヤーが飛行機に下せる命令は、移動と攻撃の2つだけです。弾が当たれば飛行機は撃墜されます。

コントロールパネルで反転しているのが命令の状態です。切り替えは、「移動」または「攻撃」の上に矢印のカーソルを合わせマウスの左側のボタンを押します。

ゲームを終了させる場合は、「終了」を選択してください。

飛行機の移動方法

- ①まずコントロールパネルで「移動」を選択します
- ②次に移動させる飛行機に矢印を合わせ左側のボタンを押します（注意：ここで絶対にマウスを動かしてはいけません。動かすとあとで泣きをみます）
- ③カーソルが矢印から鉛筆に変わります
- ④カーソルが変わったら、動かしたい方向へ思い切ってマウスを動かしてください（思い切って動かさないで移動してくれません）
- ⑤軌跡を描いて飛行機が移動します。鉛筆飛ばしの醍醐味は、先をピンピンに尖らせた鉛筆を人差し指ではじくところにあります（そのため、ときどき怪人も出ますが…）。やはりこの点を再現しないことには鉛筆飛ばしとはいえません。操作としては、移動させた位置でマウスをクリックするほうが上品で簡単なのですが、鉛筆飛ばしの雰囲気を出すためにあえてこの方法を採用しました。

マウスの本場はアメリカですから、「クリック（ボタンを押す）」とか「ドローイング（ボタンを押しながら引っ張る）」とか、いろいろなマウスのための単語があるのですが、どうやらこういう（乱暴な）使い方を考えた人がいないらしい該当する英単語がありませんでした。そこでこの動作を「プッシング」と名づけることにします。

プッシングするには広い場所が必要ですので、邪魔になるキーボードは片づけておきましょう。

移動時の注意

- ①移動した場所にはかの飛行機（敵、味方）がある場合は、空中衝突ということでも双方とも撃墜されます（特攻ですね）
- ②移動中に重なった飛行機は、うまくすり抜けたということで影響ありません
- ③ノート（表示している範囲ではなく実際の大きさ）から出てしまった飛行機は消

滅してしまいます

- ④命令が下せるのはプレイヤーの飛行機だけです。コンピュータ側の飛行機を選択すると、「これはコンピュータ側の飛行機です」と注意されます

攻撃方法

- ①まずコントロールパネルで「攻撃」を選択してください
- ②あとは「移動」と同じです。よく狙って攻撃してください
- ③弾が軌跡を描いて飛んでいきます

攻撃時の注意

- ①「移動」と違って、弾の描いた軌跡にある飛行機（敵、味方を問わず）は撃墜されます
- ②ときどきコンピュータがイカサマをして完全に当たっていても撃墜されなかったり、当たっていないのに撃墜されることがあります。これも鉛筆飛ばしによくあることですので、間違ってもコンピュータにあたらないようにしてください
- ③命令が下せるのはプレイヤーの飛行機だけです

マウスカーソルの形

状態によってマウスカーソルの形が変化します。

- ①矢印 コマンド、飛行機を選択やノートのスクロールが可能です
- ②鉛筆 移動、攻撃中です。思い切ってプッシングしてください
- ③時計 コンピュータ側の順番です。矢印になるまでお待ちください

矢印→鉛筆→時計→矢印の順に変化します。

ゲームの終了

どちらかまたは両方が全滅したらゲームエンドです。途中でやめた場合は、「終了」を選択してください。

このとき、中断した場合は、「この対戦を終了しますか？」と画面の対戦を終了するのか聞いてきます。

終了の場合は「はい」、続ける場合は「いいえ」を選択してください。

ゲームエンドの場合や画面の対戦を終了した場合は、「もう一度対戦しますか？」と聞いてきます。新たに始める場合は「はい」、プログラムを終了する場合は「いいえ」を選択してください。

ペンシルウォーズ必勝法

1 コンピュータ側の動き

「己を知り、敵を知れば百戦危うからず」の格言もあるように、相手の動きを知るのも戦いのうちです。そこでコンピュータ側の動きは次のようになっています。

- ①攻撃できる範囲に敵がいるなら攻撃する
- ②射撃はコンピュータといえども百発百中ではない
- ③原則的にはプレイヤー側の飛行機とマンツーマンで移動する
- ④動く距離は、一定の範囲内でバラバラ

そのほかの詳しい動きや射撃範囲は、実戦で修得してください。

2 勝利のための基本戦術

このゲームは、ルールが簡単でし、短時間で決着が付きまですて手軽に遊ぶことができます。しかし勝つようになるには技術が必要です。

①プッシングをマスターする

遠い相手を一発で撃墜したときの快感は当ててみた人にしかわからないでしょう。とにかくどうしてもマスターしたいのは、思ったところへプッシングできる技術。何度かやっているうちにコツがつかめると思いますので、よく練習してください。

②状況判断は的確に

ウィンドーにはノートの半分しか表示されませんが表示範囲外から攻撃されることもあります。見えない敵と戦う羽目にならないように自分と相手がチャント見える位置にノートをスクロールさせておきましょう。

③同士打ちを避ける

移動にしろ攻撃にしろ、撃墜されるときは敵味方の区別がありません。そのため、慣れないうちは同士打ちになってしまいうこともしばしばです。とくに移動で味方同士がぶつかったときは2機撃墜される（した？）ので大損害です。くれぐれも注意してください。

④落とせるときは確実に

コンピュータ側飛行機が移動の結果、思わぬところまで近づいてくれることがあります。こんなとき力が入ってしまてはいけません。力を抜いて確実に落としましょう。

⑤得意のコースを作る

慣れてくるとプレイヤーの射撃範囲はコンピュータの射撃範囲を確実に上回るハズです。そうなったら「右45°なら絶対だ!」というような得意のコースを作っておくといでしょう。「攻撃は最大の防御なり」といいますので、相手が近づいてくる前に落としましょう。

⑥一機一殺を心がける

プレイヤーとコンピュータの持ち機は同じですから、1機に何機もやられては敗北は必至です。1機やられたらやりかえすといった戦法がのまないのでしょ(といっても、これが難しいのですが...)。なるべく1機離脱といった単独行動よりも2、3機の編隊で行動するようにしましょう。

⑦いのちをけなない

鉛筆飛ばしのテクニックのひとつにイカサマがあります。コンピュータはときどきこれを使います。こんなときは熱くならず負けずので、運が悪かったと思っておきめましょう。といっても、ときどきプレイヤーの弾が相手に当たってなくても、勝手に落ちてくれることがありますので、イカサマも公平になっています。

3 これが高テクテクニックだ!

さて、それでは必殺高等テクニックを紹介しましょう。

①一石二鳥を狙え

シューティングの技術が向上してくると、長い距離を当てたり、軌跡を途中で曲げたりできるようになってきます。攻撃の場合は軌跡に重なった飛行機はすべて撃

墜れますので、射線上に複数機をのせるような位置に移動して(得意なコースにのせるといでしょう)一石二鳥を狙います。

②同士討ちを誘え

コンピュータ側の飛行機は間に味方の飛行機があっても攻撃できます。コンピュータ側の攻撃パターンをうまく利用すれば同士討ちを誘うことができます。またコンピュータ側の飛行機は移動時に目標に向かって一直線に進んでいきますので、移動線上にうまく相手の飛行機をおけば空中衝突を狙えます。

4 禁断の秘技、トラックホルスの

マウス使用法

これは邪道なのですが、どうしてもブックシングがうまくいかないときは試してみてください。

マウスはその構造上、机にボールが触れていないと動きません。これを利用して、一度マウスを浮かして、直接、指でボールを弾きます。こうすれば飛距離は格段にアップすることはいえです。

ただし、方向のコントロールは一段と難しくなりますので修練が必要です。

● プログラムについて

どうしてもマウスの飛距離を伸ばしたい場合は、

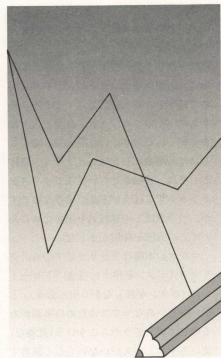
1330 MOUSE, SPEED=200

を変更してください。300~400ぐらいにすると2倍ぐらいは飛距離がアップします。しかしあまり大きくするとマウスのレスポンスが悪くなりますので注意。

88VA について

このプログラムは、姉妹誌「Oh! 98」でPC-9800シリーズ用に作ったゲームを基に88VAに移植しました。N88-日本語BASIC V3は98版のN88-BASIC (86) MS-DOS版とも互換性が高いので移植は楽でしたが、マウスやグラフィック関係などでは特有の命令を使っています。

88VAは非常におもしろいパソコンです。今回はスプライトや音楽、通信機能などは使わなかったのですが、今後これらを使って「PENCIL WARS 完璧版」を作りたいと思っています。



```
1000 *
1010 Pencil Wars
1020 *
1030 << INIT SECTION >>
1040 *START
1050 GOSUB #INIT.PROG
1060 *
1070 *RESTART
1080 GOSUB #INIT.GAME
1090 *
1100 << MAIN SECTION >>
1110 *
1120 *MAIN.LOOP
1130 WHILE EVENT1=0:WEND
1140 IF EVENT1=1 THEN GOSUB #CHECK.END:GOTO #MAIN.LOOP
1150 IF EVENT1=1 THEN *RESTART
1160 IF EVENT1=2 THEN *PROG.END
1170 END
1180 *
1190 << INIT PROGRAM >>
1200 *
1210 #INIT.PROG
1220 WIDTH 80,25:CONSOLE 1,24,0,0:DEFINT A-Z:SCREEN 2,0,0,3:OPTION BASE 0
1230 DIM UNIT(41,4),GTBL(4),GWOR(1500),MOVBL(10,2),UNITBL(41),FONT(18)
1240 SCREEN WIDTH 5,640,1:BUFFER #0,640,400,1:640,640,1:640,20,1
1250 SCREEN WINDOWS #0,(0,0),400,0:SPRITE SCREEN 1
1260 ON ERROR GOTO #ERROR.SECTION:ON STOP GOSUB #STOP.SECTION:STOP ON
1270 RETURN
1280 *
1290 << INIT GAME >>
1300 *
1310 #INIT.GAME
```

```

1320 DATA INIT
1330 MOUSE.SPEED=200
1340 EVNT1=0:X.WINDOW=8:Y.WINDOW=20:CFGL=0:MAPLINE=20
1350 RANDOMIZE VAL(MID$(DATES,1,2)):VAL(MID$(DATES,4,2)):VAL(MID$(DATES,7,2)):VAL(MID$(TIMES,1,2)):VAL(MID$(TIMES,4,2)):VAL(MID$(TIMES,7,2))
1360 FOR I=1 TO 20:UNIT(1,1)=2:UNIT(1,2)=1+20:UNIT(1,3)=INT(RND(1)*200)+10:UNIT(1,4)=100:NEXT I
1370 FOR I=21 TO 40:UNIT(1,1)=1:UNIT(1,2)=1+20:UNIT(1,3)=592-INT(RND(1)*200):UNIT(1,4)=5:NEXT I
1380 UNIT(41,1)=20:UNIT(41,2)=20
1390 SCREEN INIT
1400 GOSUB *SCREEN.CLS:BUFFER=0
1410 LINE (0,0)-(639,16),1,BF:LINE (0,16)-(639,399),0,BF,CHR$(8AA)+CHR$(8H55)
1420 LINE (X.WINDOW,Y.WINDOW)-STEP(625,368),0,BF,1:LINE (X.WINDOW,Y.WINDOW+16)-STEP(625,0),0:LINE (X.WINDOW,Y.WINDOW+337)-STEP(625,0),0
1430 FOR I=1 TO 20:CIRCLE (X.WINDOW+9,Y.WINDOW+1+16*8),4,0,...,F,1:NEXT I
1440 LINE (X.WINDOW+337,Y.WINDOW+344)-STEP(80,17),0,BF:LINE (X.WINDOW+433,Y.WINDOW+344)-STEP(80,17),0,B:LINE (X.WINDOW+529,Y.WINDOW+344)-STEP(80,17),0,B
1450 BUFFER=0:GTBL(1)=0:GTBL(2)=0:GTBL(3)=0:GTBL(4)=639:GOSUB *GPRINT:GTBL(1)=2:GPHCHR="プレイヤー軍 : 20 コンピュータ軍 : 20 ":GOSUB *GPR
1460 GTBL(1)=3:GTBL(2)=X.WINDOW:GTBL(3)=Y.WINDOW+1:GTBL(4)=X.WINDOW+625:GPHCHR="< Pencil Mars >":GOSUB *GPRINT
1470 GTBL(2)=X.WINDOW+337:GTBL(3)=Y.WINDOW+345:GTBL(4)=X.WINDOW+417:GPHCHR="移動":GOSUB *GPRINT
1480 GTBL(2)=X.WINDOW+433:GTBL(4)=X.WINDOW+513:GPHCHR="攻撃":GOSUB *GPRINT
1490 GTBL(2)=X.WINDOW+529:GTBL(4)=X.WINDOW+609:GPHCHR="終了":GOSUB *GPRINT
1500 MAP INIT
1510 BUFFER=2:RESTORE *UNIT.DATA1
1520 FOR I=0 TO 15:READ WORKS:LINE (0,1)-STEP(16,0),1,VAL("&H"+WORKS):LINE (16,15)-STEP(16,0),1,VAL("&H"+WORKS):NEXT I
1530 BUFFER=1:LINE (0,0)-(600,624),1,BF
1540 LINE (10,38)-STEP(500,0),0,8,SHAAAA:LINE (10,66)-STEP(500,0),0,8,SHAAAA:LINE (10,67)-STEP(500,0),0,8,SH80
1550 FOR I=0 TO 570 STEP 14:LINE (10,1)-STEP(500,0),0,8,SH8888:NEXT I
1560 LINE (10,54)-STEP(500,0),0,8,SH88:LINE (10,584)-STEP(500,0),0,8,SHAAAA
1570 FOR UNITNO=1 TO 40:GOSUB *PUT.UNIT:NEXT UNITNO
1580 GOSUB *PRINT.MAP
1590 MOUSE INIT
1600 MOUSE=0:RESTORE *MOUSE.AROW:GOSUB *MOUSE.CSET:MOUSE 1,1,1:ON MOUSE(2) GOSUB *MOUSE.EVENT1:MOUSE (2) ON
1610 RETURN
1620
1630 *UNIT.DATA1
1640 DATA 0000,0000,0000,0140,0140,0140,4631,5080,6143,7E3F,4141,0140,0000,0000,0000,0000
1650
1660 << GRAPHIC SECTION >>
1670 SCREEN CLEAR
1680 *SCREEN.CLS
1690 FOR I=0 TO 2:BUFFER=1:CLS 2:NEXT I
1700 RETURN
1710 GRAPHIC CHARACTER PRINT
1720 *GPRINT
1730 KANJIFLG=0
1740 IF GTBL(1)=0 THEN LINE (GTBL(2),GTBL(3))-(GTBL(4),GTBL(3)+15),1,BF,1:GOTO *GPRINT.RETURN
1750 GPXWORK=LEN(GPHCHR)*8
1760 IF GTBL(1)=1 THEN GPXWORK=GTBL(2)
1770 IF GTBL(1)=2 THEN GPXWORK=GTBL(4)-GPXWORK
1780 IF GTBL(1)=3 THEN GPXWORK=GTBL(4)-GTBL(2)-GPXWORK/2+GTBL(2)
1790 FOR GPI=1 TO LEN(GPHCHR)
1800 IF KANJIFLG=1 THEN PUT (GPHWORK,GTBL(3)),KANJI(GPXWORK+256+ASC(MID$(GPHCHR,GPI,1))):GPXWORK=GPXWORK+16:KANJIFLG=0:GOTO *GPRINT
1810 NEXT GPI
1820 GPXWORK=ASC(MID$(GPHCHR,GPI,1))
1830 IF (GPXWORK=8H1) AND (GPXWORK=8H9F) OR (GPXWORK=8H8D AND GPXWORK<8H9F) THEN KANJIFLG=1:GOTO *GPRINT.NEXT ELSE KANJIFLG=0
1840 PUT (GPXWORK,GTBL(3)),KANJI(GPXWORK):GPXWORK=GPXWORK+8
1850 *GPRINT.NEXT
1860 NEXT GPI
1870 *GPRINT.RETURN
1880 RETURN
1890 << MOUSE SECTION >>
1900 MOUSE.PATTERN SET
1910 *MOUSE.CSET
1920 MOCPS="":READ MCX,MCY:FOR MCS=1 TO 136:READ WORKS:MOCPS=MOCPS+CHR$(VAL("&H"+WORKS)):NEXT MCS:MOUSE 2,MCX,MCY,MOCPS
1930 RETURN
1940 *MOUSE.PATTERN
1950 *MOUSE.AROW
1960 DATA 00,00,20,00,20,00,01,00,01,00
1970 DATA 00,00,00,00,C0,00,00,E0,00,00,F0,00,00,F0,00,00,00,00,00
1980 DATA FC,00,00,FE,00,00,FF,00,00,F0,00,00,F0,00,00,00,00,00
1990 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2000 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2010 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2020 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2030 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2040 *MOUSE.PENCIL
2050 DATA 00,00,20,00,20,00,01,00,01,00
2060 DATA 01,00,00,00,01,00,00,00,03,00,00,03,00,00,04,40,00,00
2070 DATA 04,40,00,00,00,20,00,00,06,20,00,00,11,10,00,15,50,00
2080 DATA 15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00
2090 DATA 15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00
2100 DATA 15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00,15,50,00
2110 DATA 15,50,00,15,50,00,17,00,00,10,30,00,00,13,90,00,00
2120 DATA 00,20,00,07,C0,00,00
2130 *MOUSE.VATCH
2140 DATA 00,00,20,00,20,00,01,00,01,00
2150 DATA 07,00,00,07,C0,00,00,07,C0,00,00,0F,E0,00,00,11,10,00,00
2160 DATA 11,10,00,11,10,00,11,D0,00,00,10,10,00,10,10,00,10,10,00
2170 DATA 10,10,00,0F,E0,00,00,07,C0,00,00,07,C0,00,00,07,C0,00,00,00
2180 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2190 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2200 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
2210 DATA 00,00,00,00,00,00,00
2220 *EVENT1 LEFTPUSH (ARROW)
2230 *MOUSE.EVENT1
2240 BUFFER=0:GTBL(1)=0:GTBL(2)=X.WINDOW+8:GTBL(3)=Y.WINDOW+344:GTBL(4)=X.WINDOW+328:GOSUB *GPRINT
2250 X.MOUSE=MOUSE(4,1):Y.MOUSE=MOUSE(5,1):Y.WINDOW=MOUSE(5,1)+Y.WINDOW
2260 IF X.MOUSE<16 THEN IF (Y.MOUSE+16) AND (Y.MOUSE+368) THEN MAPLINE=(Y.MOUSE-8)/16:GOSUB *PRINT.MAP:GOTO *LEFTRETURN ELSE *LEFTRETURN
2270
2280 IF (X.MOUSE+337 AND X.MOUSE+417) AND (Y.MOUSE+344 AND Y.MOUSE+368) AND CFLG=1 THEN GOSUB *PRINT.REVERSE:GOTO *LEFTRETURN
2290 IF (X.MOUSE+433 AND X.MOUSE+513) AND (Y.MOUSE+344 AND Y.MOUSE+368) AND CFLG=0 THEN GOSUB *PRINT.REVERSE:GOTO *LEFTRETURN
2300 *NEXT1
2310 X.UNIT=X.MOUSE-17:Y.UNIT=Y.MOUSE-17*(MAPLINE-1)+16:GOSUB *CHECK.UNIT:IF HITUNIT<>0 THEN UNITNO=HITUNIT:GOSUB *PLAYER.SECTION
2320 *LEFTRETURN
2330 RETURN
2340 *EVENT2 : MOVE
2350 *MOUSE.EVENT2
2360 IF MOVNT=1 THEN MOVE1,1,MOVTL(1,1),MOVTL(1,2)
2370 MOVNT=MOVNT+1:IF MOVNT<UNITNO,4) THEN MOVE1(MOVNT,1)=MOUSE(0):MOVTL(MOVNT,2)=MOUSE(1):FOR I=1 TO MOUSE.SPEED:NEXT I:
2380 GOTO *MOVRETURN
2390 *MOUSE (1) OFF:BUFFER=0:WINDOW (X.WINDOW+17,Y.WINDOW+18)-(X.WINDOW+625,Y.WINDOW+336):VIEW (X.WINDOW+17,Y.WINDOW+18)-(X.WINDOW+625,Y.WINDOW+336)
2400 FOR MOV1=2 TO MOVNT-1:LINE (MOVTL(MOV1,1),MOVTL(MOV1,1,2))-(MOVTL(MOV1,1),MOVTL(MOV1,2)),0:NEXT MOV1

```

```

2480 WINDOW (0,0)-(639,399):VIEW (0,0)-(639,399)
2410 MFLG=1
2420 MOVRETURN
2430 RETURN
2440
2450 ' << UNIT SECTION >>
2460 UNIT PUT
2470 *PUT,UNIT
2480 IF UNIT(UNITNO,1)=1 THEN BUFFER=2:GET @(0,0)-STEP(16,16),GWORK
2490 IF UNIT(UNITNO,1)=2 THEN BUFFER=2:GET @(16,0)-STEP(16,16),GWORK
2500 BUFFER=1:PUT @(UNIT(UNITNO,2),UNIT(UNITNO,3)),GWORK:BUFFER=0
2510 RETURN
2520 UNIT REMOVE
2530 *REMOVE,UNIT
2540 GOSUB *PUT,UNIT:UNIT(UNITNO,1)=0
2550 IF UNITNO<21 THEN UNIT(41,1)=UNIT(41,1)-1 ELSE UNIT(41,2)=UNIT(41,2)-1
2560 RETURN
2570 '
2580 ' << PRINT SECTION >>
2590 INFORMATION PRINT
2600 *PRINT,INFO
2610 BUFFER=0:GTBL(1)=0:GTBL(2)=431:GTBL(3)=0:GTBL(4)=455:GOSUB *GPRINT:GTBL(1)=1:GPHCR$=STR$(UNIT(41,2)):GOSUB *GPRINT
2620 BUFFER=0:GTBL(1)=0:GTBL(2)=687:GTBL(3)=0:GTBL(4)=631:GOSUB *GPRINT:GTBL(1)=1:GPHCR$=STR$(UNIT(41,1)):GOSUB *GPRINT
2630 RETURN
2640 UNIT PRINT
2650 *PRINT,UNIT
2660 IF UNIT(UNITNO,1)=1 THEN BUFFER=2:GET @(0,0)-STEP(16,16),GWORK
2670 IF UNIT(UNITNO,1)=2 THEN BUFFER=2:GET @(16,0)-STEP(16,16),GWORK
2680 BUFFER=0:WINDOW (X.WINDOW+17,Y.WINDOW+17)-(X.WINDOW+625,Y.WINDOW+336):VIEW (X.WINDOW+17,Y.WINDOW+17)-(X.WINDOW+625,Y.WINDOW+336)
2690 BUFFER=0:PUT @(NX,UNIT-1,NY,UNIT-MAPLINE+16+15),GWORK:WINDOW (0,0)-(639,399):VIEW (0,0)-(639,399)
2700 *RETURN
2710 RETURN
2720 MAP PRINT
2730 *PRINT,MAP
2740 FOR MPI=1 TO 20
2750 IF MPI=MAPLINE THEN CIRCLE (X.WINDOW+9,Y.WINDOW+MPI+16+8),4,8,...,F,0 ELSE CIRCLE (X.WINDOW+9,Y.WINDOW+16+MPI+8),4,8,...,F,1
2760 NEXT MPI
2770 FOR MPI=MAPLINE TO MAPLINE+19
2780 BUFFER=1:GET @1,MPI+16+15)-STEP(687,15),GWORK
2790 BUFFER=0:PUT @(X.WINDOW+17,Y.WINDOW+MPI-MAPLINE+16+17),GWORK,PSET
2800 NEXT MPI
2810 RETURN
2820 <MOVE> <ATTACK> REVERSE
2830 *PRINT,REVERSE
2840 IF CFLG=0 THEN CFLG=1 ELSE CFLG=0
2850 BUFFER=0:GET @(128,0)-STEP(78,15),GWORK:PUT @(X.WINDOW+338,Y.WINDOW+345),GWORK,XOR:PUT @(X.WINDOW+434,Y.WINDOW+345),GWORK,XOR
2860 RETURN
2870 '
2880 ' << CHECK SECTION >>
2890 UNIT POINT CHECK
2900 *CHECK,UNIT
2910 HITUNIT=0
2920 FOR HITUNIT=1 TO 40
2930 IF UNIT(HITUNIT,1)<0 THEN IF UNIT(HITUNIT,2)<X,UNIT THEN IF UNIT(HITUNIT,2)+16>X,UNIT THEN IF UNIT(HITUNIT,3)<Y,UNIT THEN IF
UNIT(HITUNIT,3)+16>Y,UNIT THEN *UCEXITI
2940 NEXT HITUNIT
2950 HITUNIT=0
2960 *UCEXITI
2970 RETURN
2980 UNIT MOVE CHECK
2990 *CHECK,MOVE
3000 IF NX,UNIT<0 OR NX,UNIT>597 OR NY,UNIT<0 OR NY,UNIT>610 THEN GOSUB *REMOVE,UNIT:GOTO *MOCRETURN
3010 X,UNIT<NX,UNIT+16:Y,UNIT<NY,UNIT:GOSUB *CHECK,UNIT:IF HITUNIT<0 AND HITUNIT<UNITNO THEN *UCRASH
3020 X,UNIT<NX,UNIT+16:Y,UNIT<NY,UNIT:GOSUB *CHECK,UNIT:IF HITUNIT<0 AND HITUNIT<UNITNO THEN *UCRASH
3030 X,UNIT<NX,UNIT+16:Y,UNIT<NY,UNIT+16:GOSUB *CHECK,UNIT:IF HITUNIT<0 AND HITUNIT<UNITNO THEN *UCRASH
3040 X,UNIT<NX,UNIT+16:Y,UNIT<NY,UNIT+16:GOSUB *CHECK,UNIT:IF HITUNIT<0 AND HITUNIT<UNITNO THEN *UCRASH
3050 GOSUB *PUT,UNIT:UNIT(UNITNO,2)=NX,UNIT:UNIT(UNITNO,3)=NY,UNIT:GOSUB *PUT,UNIT:GOTO *MOCRETURN
3060 *UCRASH
3070 GOSUB *REMOVE,UNIT:SUNITNO=UNITNO:UNITNO=HITUNIT:GOSUB *REMOVE,UNIT:UNITNO=SUNITNO
3080 *MOCRETURN
3090 RETURN
3100 UNIT HIT CHECK
3110 *CHECK,HIT
3120 UNITBL(1)=0
3130 IF MOV1X<MOV2X THEN AFRM=0:GOTO *PYCHK
3140 AFRM=1:(MOV2Y-MOV1Y)/(MOV2X-MOV1X):BFRM=MOV1Y-APRM:MOV1X
3150 IF ABS(APRM)>1 THEN *PYCHK
3160 *PYCHK
3170 IF MOV1X<MOV2X THEN PAW=1 ELSE PAW=-1
3180 FOR PA12=MOV1Y TO MOV2X STEP PAW+10
3190 X,UNIT=PA12+4*Y,UNIT=INT(PA12+AFRM+BFRM)+5:GOSUB *CHECK,UNIT
3200 IF HITUNIT<0 THEN IF UNITBL(UNITBL(1)+1)<HITUNIT THEN IF HITUNIT<UNITNO THEN UNITBL(1)=UNITBL(1)+1:UNITBL(UNITBL(1)+1)=HITU
NIT
3210 NEXT PA12
3220 GOTO *PACRASH
3230 *PYCHK
3240 IF MOV1Y<MOV2Y THEN PAW=1 ELSE PAW=-1
3250 FOR PA12=MOV1Y TO MOV2Y STEP PAW+10
3260 Y,UNIT=PA12+5:IF AFRM=0 THEN Y,UNIT=MOV1X+4 ELSE X,UNIT=INT(PA12-BFRM)/APRM)+4
3270 GOSUB *CHECK,UNIT
3280 IF HITUNIT<0 THEN IF UNITBL(UNITBL(1)+1)<HITUNIT THEN IF HITUNIT<UNITNO THEN UNITBL(1)=UNITBL(1)+1:UNITBL(UNITBL(1)+1)=HITU
NIT
3290 NEXT PA12
3300 *PACRASH
3310 IF UNITBL(1)<0 THEN SUNITNO=UNITNO:FOR PA12=1 TO UNITBL(1):UNITNO=UNITBL(PA12+1):GOSUB *REMOVE,UNIT:NEXT PA12:UNITNO=SUNITNO
3320 RETURN
3330 PLAYER UNIT SEARCH
3340 *CHECK,SEARCH
3350 FOR PUNITNO=21 TO 40
3360 IF UNIT(PUNITNO,1)<0 THEN IF ABS(UNIT(PUNITNO,2)-CXPOINT)<CHLEN THEN IF ABS(UNIT(PUNITNO,3)-CYPOINT)<CHLEN THEN *PSEXIT
3370 NEXT PUNITNO
3380 PUNITNO=0
3390 *PSEXIT
3400 RETURN
3410
3420 ' << PLAYER SECTION >>
3430 ' PLAYER SECTION
3440 *PLAYER,SECTION
3450 IF UNIT(UNITNO,1) THEN GTBL(1)=1:GTBL(2)=X.WINDOW+8:GTBL(3)=Y.WINDOW+344:GTBL(4)=X.WINDOW+328:GPHCR$="それはコンピュータ側の飛行機です":GOSUB
*GPRINT:GOTO *PARETURN
3460 RESTORE *MOUSE,PENCIL:GOSUB *MOUSE,CSET
3470 MFLG=0:MOCKNT=1:MOVTBL(1)=UNIT(UNITNO,2)+X.WINDOW+21:MOVTBL(1,2)=UNIT(UNITNO,3)+Y.WINDOW+38-MAPLINE+16
3480 ON MOUSE(1) GOSUB *MOUSE,EVENT2:MOUSE (1) ON
3490 WHILE MFLG=0:WEND

```



```

3500 MOUSE (1) OFF:ON MOUSE(2) GOSUB *MOUSE.EVENT1
3510 IF CFLG=0 THEN GOSUB *PLAYER.MOVE ELSE GOSUB *PLAYER.ATTACK
3520 GOSUB *COMPUTER.SECTION
3530 *PARRETURN
3540 GOSUB *PRINT.MAP:GOSUB *PRINT.INFO
3550 IF UNIT(41,1)<1 OR UNIT(41,2)<1 THEN EVENT1=-1
3560 RETURN
3570
3580 *PLAYER.MOVE
3590 NX,UNIT=MOVTBL(UNIT(UNITNO,4),1)-X,WINDOW-22:NY,UNIT=MOVTBL(UNIT(UNITNO,4),2)-Y,WINDOW-37:MAPLINE=16
3600 GOSUB *PRINT.UNIT:GOSUB *CHECK.MOVE
3610 RETURN
3620
3630 *PLAYER.ATTACK
3640 FOR PAI=2 TO UNIT(UNITNO,4)
3650 MOVIX=MOVTBL(PAI-1,1)-X,WINDOW-22:MOVIY=MOVTBL(PAI-1,2)-Y,WINDOW-37:MAPLINE=16
3660 MOVY=MOVTBL(PAI-1,1)-X,WINDOW-22:MOVY2=MOVTBL(PAI,2)-Y,WINDOW-37:MAPLINE=16
3670 GOSUB *CHECK.HIT
3680 NEXT PAI
3690 RETURN
3700
3710 << computer section >>
3720 *COMPUTER SECTION
3730 *COMPUTER SECTION
3740 PSUNIT=UNITNO:RESTORE *MOUSE.WATCH:GOSUB *MOUSE.CSET
3750 FOR UNITNO=1 TO 20
3760 IF UNIT(UNITNO,1)<0 THEN CXPOINT=UNIT(UNITNO,2):CYPOINT=UNIT(UNITNO,3):CHLEN=UNIT(UNITNO,4):GOSUB *CHECK.SERCH:IF PSUNIT<>0 THEN
N GOSUB *COMPUTER.ATTACK:GOTO *CORETURN
3770 NEXT UNITNO
3780 *COMPUTER MOVE
3790
3800 UNITNO=PSUNITNO-20:IF UNIT(UNITNO,1)=0 THEN FOR UNITNO=1 TO 20:IF UNIT(UNITNO,1)=0 THEN NEXT UNITNO
3810 IF UNIT(PSUNITNO,1)=0 THEN FOR PSUNITNO=21 TO 40:IF UNIT(PSUNITNO,1)=0 THEN NEXT PSUNITNO:GOTO *CORETURN
3820 CXPOINT=UNIT(UNITNO,2):CYPOINT=UNIT(UNITNO,3):CHLEN=UNIT(UNITNO,4)
3830 CXPOINT=UNIT(UNITNO,2):CYPOINT=UNIT(UNITNO,3):CHLEN=UNIT(UNITNO,4)
3840 IF CXPOINT=PXPOINT THEN CYPOINT=CYPOINT+SGN(PYPOINT-CYPOINT):INT(CHLEN+RND(1)):GOTO *CMXEXIT
3850 IF CYPOINT=PYPOINT THEN CXPOINT=CXPOINT+SGN(PXPOINT-CXPOINT):INT(CHLEN+RND(1)):GOTO *CMXEXIT
3860 APRMI=(CYPOINT-PYPOINT)/(CXPOINT-PXPOINT):BPRMI=CYPOINT-APRMI*CXPOINT
3870 IF ABS(APRMI)<1 THEN CXPOINT=CXPOINT+SGN(PXPOINT-CXPOINT):INT(CHLEN+RND(1)):CYPOINT=CINT(APRMI)*CSNG(CXPOINT+BPRMI) ELSE CYPOINT=
CYPOINT+SGN(PYPOINT-CYPOINT):INT(CHLEN+RND(1)):CXPOINT=CINT(1-CSNG(CYPOINT+BPRMI)/APRMI)
3880 *CMXEXIT
3890 NX,UNIT=CXPOINT:NY,UNIT=CYPOINT
3900 BUFFER=0:WINDOW (X,WINDOW+17,Y,WINDOW+18)-(X,WINDOW+625,Y,WINDOW+336):VIEW (X,WINDOW+17,Y,WINDOW+18)-(X,WINDOW+625,Y,WINDOW+336)
3910 LINE (UNIT(UNITNO,2)+X,WINDOW+22,UNIT(UNITNO,3)+Y,WINDOW+37:MAPLINE=16)-(CXPOINT,X,WINDOW+22,CYPOINT,Y,WINDOW+37:MAPLINE=16),0
3920 WINDOW (0,0)-(639,399):VIEW (0,0)-(639,399)
3930 GOSUB *PRINT.UNIT:GOSUB *CHECK.MOVE:GOTO *CORETURN
3940 *CORETURN
3950 RESTORE *MOUSE.ARROW:GOSUB *MOUSE.CSET
3960 RETURN
3970
3980 *COMPUTER ATTACK
3990 UNWORK=12*INT(UNIT(41,1)/10)+10:MOVIX=CXPOINT:MOVIY=CYPOINT:MOVX2=UNIT(PSUNITNO,2)+INT(RND(1)*UNWORK)-INT(UNWORK/2):MOVY2=UNIT(PUN
ITNO,3)+INT(RND(1)*UNWORK)-INT(UNWORK/2)
4000 PXPOINT=MOVX2-X,WINDOW+22:PYPOINT=MOVY2+Y,WINDOW+37:MAPLINE=16
4010 CXPOINT=PXPOINT+X,WINDOW+22:CYPOINT=PYPOINT+Y,WINDOW+37:MAPLINE=16
4020 BUFFER=0:WINDOW (X,WINDOW+17,Y,WINDOW+18)-(X,WINDOW+625,Y,WINDOW+336):VIEW (X,WINDOW+17,Y,WINDOW+18)-(X,WINDOW+625,Y,WINDOW+336)
4030 LINE (CXPOINT,CYPOINT)-(CXPOINT,PYPOINT),0
4040 WINDOW (0,0)-(639,399):VIEW (0,0)-(639,399)
4050 GOSUB *CHECK.HIT:GOSUB *PRINT.MAP
4060 RETURN
4070
4080 << GAME OVER SECTION >>
4090
4100 *CHECK_END
4110 MOUSE (2) OFF:BUFFER=0:GET @(120,0)-STEP(78,15),GWORK:PUT @(X,WINDOW+530,Y,WINDOW+345),GWORK,XOR
4120 BUFFER=0:LINE (100,100)-STEP(440,120),0,BF,1
4130 GTBL(1)=3:GTBL(2)=100:GTBL(3)=110:GTBL(4)=540:GPCHR0="プレイヤー軍 "+STR$(UNIT(41,2))" コンピューター軍 "+STR$(UNIT(41,1)):GOSUB *GP
RINT
4140 IF (UNIT(41,1)>0 AND UNIT(41,2)>0) OR (UNIT(41,1)<0 AND UNIT(41,2)<0) THEN GPCR0="*** 引き分け ***" ELSE IF UNIT(41,1)<0 THEN G
PCR0="*** プレイヤー軍勝利 ***" ELSE GPCR0="*** コンピューター軍勝利 ***"
4150 GTBL(1)=3:GTBL(2)=100:GTBL(3)=110:GTBL(4)=540:GOSUB *GPPOINT
4160 ON MOUSE(2) GOSUB *MOUSE.EVENT4
4170 IF UNIT(41,1)>0 AND UNIT(41,2)>0 THEN GPCR0="この対戦を終りますか?":GOSUB *YESNO:IF EVENT4=2 THEN LINE (100,100)-STEP(440,120),1,BF
*GOSUB *PRINT.MAP:EVENT1=0:GOTO *PARRETURN
4180 GPCR0="もう一度、対戦しますか?":GOSUB *YESNO:EVENT1=EVENT4
4190 *PARRETURN
4200 ON MOUSE(2) GOSUB *MOUSE.EVENT1
4210 GET @(120,0)-STEP(78,15),GWORK:PUT @(X,WINDOW+530,Y,WINDOW+345),GWORK,XOR
4220 MOUSE (2) ON
4230 RETURN
4240
4250 YESNO
4260 YESNO
4270 EVENT4=0
4280 BUFFER=0:GTBL(1)=0:GTBL(2)=101:GTBL(3)=162:GTBL(4)=539:GOSUB *GPPOINT:GTBL(1)=3:GOSUB *GPPOINT
4290 LINE (224,100)-STEP(80,17),0,BF,1:LINE (336,100)-STEP(80,17),0,BF,1
4300 GTBL(1)=3:GTBL(2)=224:GTBL(3)=100:GTBL(4)=304:GPCHR0="はい":GOSUB *GPPOINT
4310 GTBL(2)=336:GTBL(4)=416:GPCHR0="いいえ":GOSUB *GPPOINT
4320 MOUSE (2) ON
4330 WHILE EVENT4=0:WEND
4340 MOUSE (2) OFF
4340 BUFFER=0:GET @(120,0)-STEP(78,15),GWORK:PUT @(EVENT4+1)*112+1,189),GWORK,XOR
4350 RETURN
4360
4370 *MOUSE.EVENT4
4380 X,MOUSE=MOUSE(4,1):Y,MOUSE=MOUSE(5,1)
4390 IF (X,MOUSE)>224 AND X,MOUSE<304 AND (Y,MOUSE)>100 AND Y,MOUSE<204) THEN EVENT4=1
4400 IF (X,MOUSE)>336 AND X,MOUSE<416 AND (Y,MOUSE)>188 AND Y,MOUSE<284) THEN EVENT4=2
4410 RETURN
4420
4430 << END SECTION >>
4440
4450 *PROG.END
4460 MOUSE 0:GOSUB *SCREEN.CLS
4470 RETURN
4480
4490 << STOP SECTION >>
4500
4510 *STOP SECTION
4520 RETURN
4530
4540 << ERROR SECTION >>
4550
4560 *ERROR SECTION
4570 RESUME NEXT

```

パソコン通信が普及するにつれ、各メーカーよりリーズナブルなものが発売されるようになった。この年で20,000円台のモデムが相次いで出てきた。デザイン的にもスマートなものが増え、性能も各メーカーによりさまざまである。今回は、ユーザーに身近になったモデムのカタログをお送りする。

1200BPSモデム

通信速度1200BPS (BIT/sec) のモデムは300BPSにも対応していると考えてもらって差し支えない。

- MD1200E (オムロン) 24,800円



通信方式/全二重 NCU形式/AA 通信規格/CCITT V21/22, BELL 103/212A コマンド/ヘイズAT, CCITT V25bis 外寸法/168(W)×250(D)×37.5(H)(mm), 重量/約500g セルフチェック機能搭載

- PV-A1200Mk II (AIWA) 26,800円



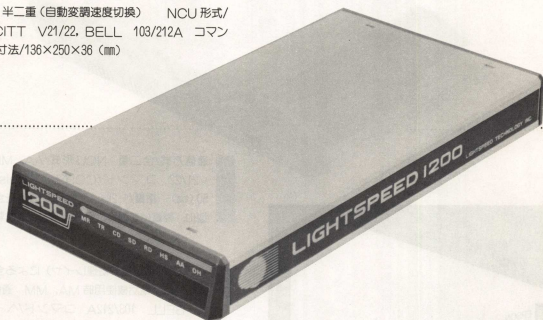
通用回線/電話回線, PBXの回線 通信方式/全二重 NCU形式/AA, MA, MM 通信規格/CCITT V21/22, BELL 103/212A コマンド/ヘイズAT 外寸法/160×222×47.5 (mm) 重量/1.2kg ACアダプタ内蔵により電源の安定化, および接触不良による事故を未然に解消

モデムについて知らないという人はもういないと思うが、ここで簡単に説明をしよう。モデムとはパソコンとパソコンを電話回線を使ってつなぐためのもの。

パソコンの裏側についているRS-232C端子にモデムをつなぐと、RS-232Cから出てくるデジタル信号(直流信号)を電話回線用のアナログ信号(交流信号)に換え(変調)、電話回線に出力する。また電話回線により入ってきたアナログ信号をコンピュータ用のデジタル信号に換えて(復調)、コンピュータに出力する。このモデムも最近では非常に安くなってきており、一般ユーザーにも手の届く価格になってきた。プロトコルの充実によって通信エラーも少なくなってきている。

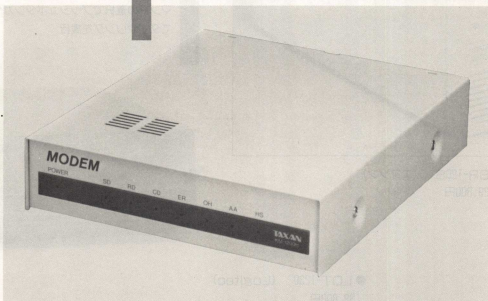
●LIGHT SPEED1200 (昭和システム研究所)
18,800円

通信方式/全二重, 半二重(自動変調速度切換) NCU形式/
AA 通信規格/CCITT V21/22, BELL 103/212A コマン
ド/ヘイズAT 外寸法/136×250×36 (mm)



●ハロハロ KM-1200H (TAXAN)
29,800円

通信方式/全二重 NCU形式/AA 通信規格/CCITT V21/22
コマンド/ヘイズAT 外寸法/174×210×46 (mm) 重量/約1.5
kg 自動通話切り換え機能でパソコンにつなげだま電話使用可能



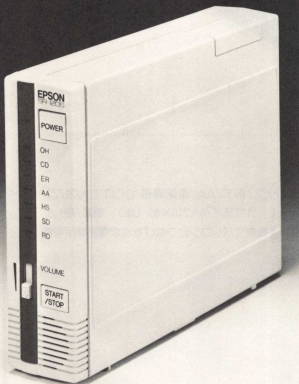
● COMSTER 1212 (NEC)
43,800円



通信方式/全二重 NCU形式/AA, MM 通信規格/CCITT V21/22 コマンド/CCITT V25bis 外寸法/160×285×50 (mm) 重量/1.2kg 高集積度 LSI を全面的に使用し大幅な小型化, 軽量化を図り, 低消費電力と高信頼度を実現

通信方式/全二重 (物理レイヤ) による全/半二重 NCU 形式/AA 内蔵, 電話機使用時 MA, MM 通信規格/CCITT V21/22, BELL 103/212A コマンド/ヘイズ AT 外寸法/44×200×140 (mm) 重量/0.5kg コンパクトな外づけの縦置きタイプ

通信方式/全二重 NCU形式/AA, MM 通信規格/CCITT V21/22, BELL 103/212A コマンド/ヘイズ AT 外寸法/220×250×78 (mm) 重量/約 2 kg タイヤリング動作においてコマンドの選択でプッシュボタンかタイヤリング/バリスカを判断してタイヤリングを実行



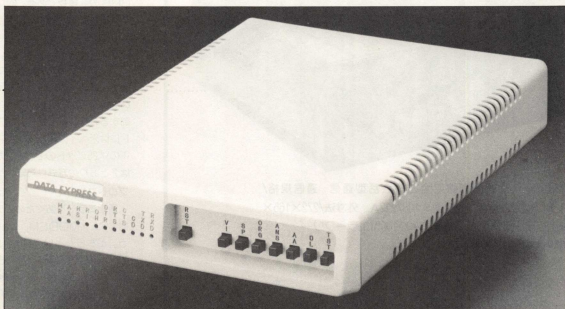
● SR-120S (エプソン)
29,800円



● LDT-1230 (Logitec)
39,800円

2400BPSモデム

通信速度は2400, 1200, 300BPSである。



- DATA EXPRESS2400
(ベクトル電子)
89,800円

通信方式/全二重 NCU形式/AA, MA, MM 通信規格/
CCITT V21/22/22bis, BELL 103, 212A コマンド/ハイ
スAT 外寸法/203×280×45 (mm) MNP エラーコントローラ
(オプション) 搭載可能 (MNPはマイクロコム社の登録商標)



- PV-A2400 (AIWA)
49,800円

通信方式/全二重 NCU形式/AA, MA, MM 通信規格/
CCITT V21/22/22bis, BELL 103/212A コマンド/ハイ
スAT 外寸法/160×222×47.5 (mm) 重量/1.4kg これはP
V-A1200Mk II の上位機種

● MICROCOM AX/2400J

(アメリカン・テクノロジー・グループ)

149,000円



通信方式/半二重通信回線で全二重会話型通信 通信規格/
CCITT X22/22bis, BELL 103/212A 外寸法/272×165×
46 (mm) テータチェック機能により100%エラーフリー

(注：写真は同シリーズのAX19624Jです)

● COMSTAR 2424

(NEC)

128,000円



通信方式/全二重 NCU形式/AA, MA, MM 通信速度/
1200, 2400BPS 外寸法/160×285×50 (mm) 重量/1.4kg
MNP 標準装備で相手側に MNP 機能がないモデムに対しては
それを自動的に判断して通信を行う

通信速度

送るデータ量が等しく通信速度が速くなれば逆間数的に伝送時間は減ってくる。伝送時間が減ればそれだけ伝送料金は減る。その代わり速いものは高価ではある。モデムの通信速度は BPS (ビット/秒) という単位で表され300BPS, 1200BPS, 2400BPS の3種類である。同じ通信方式では300BPSのそれぞれ4倍, 8倍速いことになる。

コマンド

インテリジェントモデムには電話をかけるためのダイヤルやプッシュホンはついていない。電話をかけるときにはNCUを制御するコマンドを使う。NCU制御コマンドには米ヘイズ社のATコマンドとCCITTのV.25ビットの2とおりがある。最近では2とおりの方式に対応したモデムも多い。

プロトコル

通信エラーを防ぐためのチェック方式でMNP, XMODEM, Kermit などがある。どちらかといえばMNPはアメリカで、XMODEMは日本で多く使われている。チェックはデータを一塊にし、そのデータの四則演算の結果を照らし合わせるという方法が一般的である。ものによってはエラーがなければ一塊を大きくして伝送時間を短縮するものもある。しかしエラーを完全に防げるわけではない。

全二重/半二重方式

全二重とは、送信と受信が同時に行える方式で、半二重とは同時に行えない方式である。最新のモデムは全二重方式も使えるものがほとんどである。

NCUモード (AA, MM, MA)

NCUには、AA型とMA型、MM型の3種類の方式がある。かかってきた電話を自動的にとり、コンピュータにつなぐことを自動着信、コンピュータにより電話をかけることを自動発信という。自動着発信型をAA型、自動着信型をMA型、自動発信型をMM型という。最近のインテリジェントモデムのほとんどがAA型である。

電話回線

電話回線には2種類のダイヤル回線とプッシュ回線の3種類がある。これは手元に引いてきている回線の種類で、モデムはダイヤルスイッチなどによって切り替えられるようになっている。これはモデムの性能などにはまったく関係ないものと考えてもらって構わない。しかし相手側がプッシュホンの音に反応して作動するものもあるので、プッシュホンの方が使い勝手はよいだろう。

VOW WOW

●読者のページ

本誌もめでたく2号めを迎えることができました。読者のみなさん、応援してくれてありがとう！ 創刊号のアンケート葉書きは、はげましのお便りでいっぱい。これからいろんなお便り、お待ちしております。

V Aばんざい！

VAっていいですね。今まであまり関心がなかったのですが、「アニメフレーマー」やV3モードの記事を読むと、うらやましくなりました。ちょっと高いけど…。

岩国市・原田安規・17歳

VAの記事がよくまとめたあつたと思います。ただ、V3BASICについてもっと詳しくしてほしい。VA中心！ 月刊にして！ 980円高い！

鈴鹿市・尾藤正明・16歳

最近 X68000が伸びてきて、VAは押され気味ですがこの特集を読んで自信ができました。パソコンは「機能」じゃなくて「どう使うか」ですよ。臨時増刊なんて言わずに、月刊にしてください。

全国の88ユーザーは出してくれないと毎晩フンのなかで泣きます！

福岡市・花田伸一・15歳

プログラムは実用性があり大変いと思います。しかし僕のようにパソコンを買ってすぐ受験という人間には打ち込んでいる暇などなく、マシン語もサッパリ。ソフトとして読者に売ってほしいなあ。

宇都宮市・大塚剛伯・15歳

VAがはじくなりました。私がFHを買った1か月後にVAが出たのです。今思えばもう少し待っていればVAを買えたの

に…（うー腹立つなー）。

東大阪市・森田 準・25歳

VAの記事なんて今さら…という気がしました。はっきり言って、僕らのアンテナはもうVAの次の機種に入ってるのです。

いわき市・小野浩文・20歳

特集が88VAだったので本誌を購入しました。私は88VAを持っているのですが買ったとき、まだVA用のソフトもVA対応のソフトも少なくながっかりしていましたが、この特集を読んでやっぱり良かったと思っています。

山武郡・谷本 忠・32歳

88VA専用のプログラムの「お絵描きくん」を見てその行数の少なさに驚きました。

大阪市・安富 良・13歳

88VAは、それに関する記事を読めば読むほど素晴らしいハードであるように思えます。心の底から欲しいと思います。とくに、日本語処理の素晴らしいに驚きました。

とりあえずVAを買う予定はないので（欲しいけど）VA以前の88でも参考になるような物がよいな、と思います。

国分寺市・宮坂 肇・18歳

うーVAめー、うーさには聞いていたが、奥をじっくりかえてみると、とてももてないはないか。VAが欲しいよー。でもMy SR チャンもかわいがってあげなくては…。

小牧市・舟橋健司・17歳

待ってました！ とばかりに買ってきたが、臨時増刊のわりには内容がいささか乏しく、価格も高いと思いました。だけど、もっと早く出版しても良かったと思います。

気仙沼市・上田克朗・25歳

みんな企画 どうでしょう？

ミュージック・インタフェースボードの制作やMIDI楽器をつないだ演奏プログラム、PSG音源の音色の作り方など88を使ったコンピュータ・ミュージックの特集をやってほしいです。

多賀城市・佐藤和広・20歳

88にはこういうアプリケーション雑誌が少ないのでとても便利です。もっと便利で使いやすいツールを発表してください。デイスアセンブラなどいいなあ。

明石市・柿原正朗・14歳

記事はVA、プログラムは88用となっていて、もう少し88用の記事を載せて欲しいかったですね。でも、プログラムはとてもよいものだったと思います。

札幌市・松原 勇・18歳

ゲームプログラムとコンピュータ・ミュージックの特集を多数掲載されることを期待します。それから、マシン語モニタについて解説して欲しいと思います。

京都市・西田知弘・19歳

88シリーズの機能をフルに生かせるプログラムや、隠れた機能を生かせるプログラムをお願いします。それからプログラムの入力方法も書いて欲しいかと思っています。

佐用郡・古川茂樹・31歳

初心者向けの質問コーナーやマシン語の打ち込み方法など、いくらマニュアルを熟読したところでもわからないことを掲載してほしいと思います。

名古屋市・羽場推支・32歳

今月のP.1のような新製品の紹介は便利です。アプリケーションはあればあるほどよいと思います。今月のN88-BASICの軌跡みたいにしみじみしてしまうような記事もうれしいですね。

新田郡・高橋陽一・19歳

僕のもそうだけどPC-8801が眠っているので88がバツと生かせるような記事を！

羽島市・服部春男・28歳

8801を見捨ててください。近頃ではゲームもほとんどSR以降用だし、私はかない。

静岡市・斉藤俊久・28歳

DOSの制作、グラフィック・ツールの作り方、マシン語による周辺機器の操作、一応はマシン語を覚えたい人のための応用的なおこなをお願いします。

吹田市・尾崎秀峰・16歳

各ソフトのレポート、外字の作成の仕方、外字の色々な例などをおねがいします。

寝屋川市・大賀保巳・33歳

VAのグラフィック、ミュージックなどのAVについてのことを多く取り上げて欲しいとおもいます。

可見市・中沢 博・16歳

編集室より

前号のVAの特集、いかがでしたか？みなさんからたくさんのお便りをいただきました。どうもありがとう。今回88の新シリーズ情報をお届けしましたが、これからますます88は次なる世界を広げ、私たちユーザーをドキドキさせてくれるでしょう。本誌ではこれからも88を追究し、同時に全88ユーザーをサポートしていくつもりです。どうぞヨロシク！

88プレゼント

1

①色まで黒、これこそその名のとおり「マウス」。和知電子より4名に。

2

②オーディオラボより本誌でも紹介したサンプリング用CD「SOUND PACK 128」を3名とステッカーを10名に。

3

③スタジオBCより「CG ディスク」プレゼント

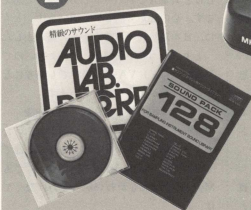
本誌14ページから16ページのカラーページで紹介したCGをディスクにして、計5名にプレゼント。

ご希望の方はアンケート葉書きに希望のプレゼント番号を1つ明記してください。発表は本誌3号（4月28日発売予定）に行います。

1



2



● Oh!88 記事名

● ご住所

↓ 返信用アて名シールになります。楷書でお書きください。

● 質問事項

エラーが出る場合には、プログラムの番号、行番号、エラーメッセージを明記してください。質問は簡潔にお書きください。

● 掲載記事についてのお問い合わせ先

〒102 東京都千代田区九段南 2-3-26 井間ビル
株式会社バンク出版事業部 Oh!88 質問係

● 応募要領

上記あてに、質問用紙、および返送先を明記して60円切手を貼った定形封筒を同封してください。

● 質問用紙の取り扱い

お送りいただいた質問用紙は、係から作者あてに転送します。ただし、下記の注意が守られていない場合、ご返事でかかれますので、あらかじめご了承ください。また郵送のため、回答に時間がかかります。

● 応募上の注意

・バグについては月曜日から金曜日（年末年始、夏期休業中を除く）の午後4時から6時まで、P.160記載のバグ電話にてお受けします。あらかじめ電話でバグの有無についてご確認ください。この質問用紙では、バグ以外のトラブルについてサポートします。ただし、

①初歩的な打ち込みミス（Syntax Error, Illegal Function Call, Subscript Out of Range, Wend without While など）の場合は、もういちど打ち込みをご確認ください。

②対応機種、対応プリンタの拡張、機能の変更については作者への「ご要望」としてバージョンアップの参考とさせていただきますが、個別の回答は致しません。

③マシンの語の打ち込み方法、DISKBASIC、MS-DOSの取り扱いなど、初歩的な内容、マニュアルを熟読すればご理解いただける内容についてのご質問はご遠慮ください。

④ご質問は、P.160記載のバグ電話もしくは質問用紙にてお願いします。作者に直接のご質問はできません。

⑤本誌発売後1年以上経過した場合、作者の都合により、バグの訂正以外の質問にお答えできない可能性があります。

Oh!88 | N F O R M A T I O N

「Oh!98」の姉妹誌として出発しました「Oh!88」、いかがでしたか？ 今後も PC-8800 シリーズユーザーのために、耳よりな情報やお得で役立つプログラムを満載します。乞うご期待！

そこで「Oh!88」編集部では、読者の方からのオリジナルプログラムを募集します。応募の方法は以下のとおりです。

① PC-8800 シリーズ上で実行可能な自作プログラム。長さ、使用言語は自由。

また、グラフィックツールを使ったイラスト、マンガ、CG、音楽、ビデオなど自作のものなら何でも

OK！

② 応募プログラムと内容を説明した原稿に対応マシン、メディア、必要な周辺機器を書き、住所、氏名、年齢、職業、連絡先 TEL を明記してください。

なお、プログラムはディスクとプリンタ出力したリスト、両方お送りください。

③ 優秀なプログラムは本誌に採用させていただきます。掲載につきましては、本誌規定の原稿料をお支払いいたします。

では、ユニークなプログラムをお待ちしています！

編集後記

■PC-8800シリーズをたっぷり使いこなし、という願いをこめて第2号をお送りします。もはや8ビットの時代ではないという乱暴な議論がありますが、これからのパソコン世代を担うのは、8ビットマシンを使いこなす人々の手にかかるころが大きいはず。その面でOh!88がお役に立てばこの雑誌を送り出すものにとって非常に幸いだと思えます。本年も8ビットマシンも面白くしましょう。(G)

■私は「アガリ性」です。郷里の徳島には高級レストランも外人も皆無で、きらびやかな場所、洗練されたレストラン、外人さんがいっぱいいるところに行くと異常に緊張してガチガチになってしまいます…と同郷の漫画家・柴門ふみさんが言っていました。田舎モンゆえしょうがない。かえって最近東京と田舎のギャップをひしひしと感じ、田舎の方が住みやすいなあ、なんて感傷にふける今日この頃。(ZUWAI)

■今年最初のコンピュータどっかんニュース。88年が明けた途端に世界中のSUN3 (UNIXワークステーションマシンのベストセラー) が、いきなりドッカンドッカンと暴走を始めて、あちらこちらの研究所やオフィスで上や下への大騒ぎ。原因はシステムプログラムのうろうろ処理のバグだったそう。とにかくドッカンは怖い。うちのマシンはだいじょうぶかしらん。今年1年、何事もなく無事過ごせすように。(竹)

■今回のOh!88の仕事は、前回よりも楽だった。とはいえ、暮の31日から新年7日まで、プログラムにはまってしまう(といいつつゲーセン行ってアフターバーナーも練習したが。コンスタント12ステージオーバーってとこな)。それにしても、88VAってすこい。X68000なんか問題じゃない。あんなゲームがBASICでできるなんて。アフターバーナーやアウトランもできないかなあ。(匿名希望あぼ)

プログラム・バグのお問い合わせは月曜日
から金曜日の午後4時から6時の間、右記
の電話にてお受けいたしております。

03-237-8667

臨時増刊 Oh!88 No.2

■1988年2月10日発行 第6巻第16号通巻85号

■定価 980円

■発行人 ●孫 正義

■編集人 ●山口幸男

■発行元 ●株式会社日本ソフトバンク 出版事業部

■営業部 ●東京都千代田区九段南2-3-26 井間ビル 〒102

TEL 03-261-4095 FAX 03-262-8397

■編集室 ●東京都千代田区九段南2-3-26 井間ビル 〒102

TEL 03-237-7895 FAX 03-263-9754

■本社 ●東京都千代田区九段南2-3-14 靖国九段南ビル2F 〒102

TEL 03-263-3690 FAX 03-263-3660

■西日本営業部 ●大田区東区南町2-6 明治生命ビル10F

〒541 TEL 06-264-1471

■印刷 ●凸版印刷株式会社

©1988 SOFT BANK CORP 雑誌02182-2

本誌からの無断転載、および著作権者の権利を侵害する行為を禁じます。

●編集長 橋本五郎

●編集 塚谷典子

●協力スタッフ 竹田津 恩 大島 萬 佐藤 淳

●校正 グループこじら

●写真 杉山和典

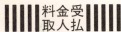
●AD 阿部孝治 (スタジオ203)

●表紙イラスト 藤沢リュウジ

●レイアウト ペンシルハウス

●広告 菊地実宅

●出版営業 中藤文男



料金受
取人払

郵便はがき

1 0 2

麴町局
承認

7120

差出有効期間
昭和64年 9 月
7 日まで

東京都千代田区九段南
2-3-26井関ビル

(株)日本ソフトバンク

Oh!88 係行

フリガナ
●お名前

男 ・ 女

歳

●おところ 〒

●TEL

●ご職業・勤務先

●学校・学部・学年

●購読誌

●主に使うOSおよび言語

愛読者

カード

No.2

プレゼント

番

●本誌の特集を読んで、いかがでしたか

●良かった記事(特集以外で)

1

2

3

●つまらなかった記事(特集以外で)

1

2

3

●今後Oh!88でとりあげてほしい企画・特集

●お持ちのシステムと主なソフト

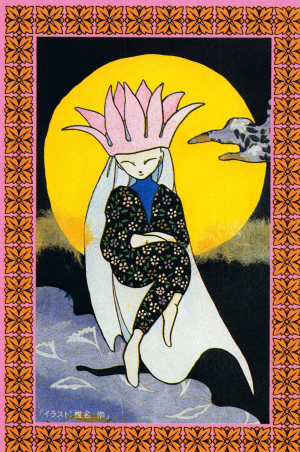
ハード

PC-8801

ソフト

●パソコン歴

神様が、へん。



愛と夢、そして哀しみのアクティブ・アドベンチャーゲーム

かわいそう物語

PRESENTED
BY
Tiny/Tiny to the Moon/Fractal TINYAN/ルー
PC-88VA専用

その頃、世界は7つの国から成り、それぞれを治めていた7人の神様がいた。ある日のこと、朝から妙な音が1日じゅう鳴り響いた。それからだ。神様が変わったのは。人びとは苦しめられ、嘆き、おびえていた。やがて7人の若者が「神様を倒そう」と立ち上がったのだが、果たして……。「かわいそう物語」は、PC-88VA専用版

のアクティブ・アドベンチャーゲームです。個性的な主人公たちとドラマティックなストーリーで、冒険はいつでも楽しくなります。

〈好評発売中〉

- PC-88VA ■5"-2HD(2枚組)
 - 640×400ドット・アナログRGBディスプレイが必要です。
 - PC-88VA添付のマウスを使用します。
- 定価 7,800円



MA/FA

こころに響くサウンド機能で、
イマジネーションの境界線を
いっきに跳び超えろ。



新登場

PC-8801MA NECパーソナルシステムPC-8801シリーズ
1Mバイトタイプ・インポート・ビデオ2枚内蔵
(320Kバイトタイプの読み取り/書き込みも可能)
本体標準価格.....198,000円

PC-8801FA NECパーソナルシステムPC-8801シリーズ
320Kバイトタイプ・インポート・ビデオ2枚内蔵
本体標準価格.....168,000円

●FM6音、リズム6音の迫力あるステレオ12音源。●クラス初、ADP
OMによるデジタルサンプリング機能を搭載。●オリジナルなサウンド
が簡単に創れるソフト“インスタントミュージック”(別売)の提供。●辞書
ROM搭載による連文節変換[®]などの高度な日本語処理機能。[※]PC-8801
MA/FA ●PC-8801シリーズの豊富なソフトウェア、周辺機器が活用可能。



斉藤由貴



聴こえるよわたしの音色。

■お問い合わせは、最寄りのNECへ

北海道支社(札幌) 011(25)15531
中部支社(名古屋) 052(283)3611
中国支社(広島) 082(247)4111

東北支社(仙台) 022(26)12551
北陸支社(金沢) 0762(23)1621
四国支社(高松) 0878(22)4141

東京支社(東京) 03(455)3111
関西支社(大阪) 06(231)3111
九州支社(福岡) 092(271)7700

■技術的なご質問へ直接に電話でお答えします。

NECパソコンインフォメーションセンター 東京 03(452)8000 大阪 06(211)9800
受付時間:9:00-17:00 月曜日-金曜日(祝日を除く)電話番号は、よくお確かめのうえおかけください。

NECのパソコンファミリー
日本電気グループ

雑誌02182-2 定価980円

2

100億ユーロの
 プリカーション・
 プラックの
 特集
 大
 大
 大

F.A.M.A. V.A.

[illegible]